

A DESIGN AND IMPLEMENTATION METHODOLOGY

FOR LAND INFORMATION SYSTEMS

A CASE STUDY

by

Rutherford

William R. Love (B.Surv Hons)

Graduating 1986

Submitted in partial fulfilment
of the requirements for the degree of

Master of Surveying Science

UNIVERSITY OF TASMANIA

HOBART

October 1985

Except as stated herein this thesis does not contain any material which has been accepted for the award of any other degree or diploma in any university nor, to the best of my knowledge and belief, does it contain any copy or paraphrase of material previously published or written by another person, except where due reference is made in the text of the thesis.

Signed:

A handwritten signature in dark ink, appearing to be "W. H. R. Jones", written over a horizontal line.

CONTENTS

INTRODUCTION	(i)
Chapter 1. SYSTEMS DESIGN AND DEVELOPMENT	1
1.1 Introduction	1
1.2 Traditional Approach to Systems Design	2
1.3 Deficiencies in the Traditional Approach	3
1.4 Improving the Traditional Approach	5
1.5 Conclusion	8
Chapter 2. THE DESIGN AND DEVELOPMENT OF A LAND INFORMATION SYSTEM	
2.1 Introduction	9
2.2 Project Environment	10
2.3 Description of the Actors	11
2.4 Selection of a Systems Development Life-Cycle (SDLC)	12
2.5 Implementation of the SDLC	15
2.5.1 Implementation Plan	17
2.5.2 Familiarisation	17
2.5.3 System Definition	19
2.5.3.1 Requirements Formulation and Analysis	19
2.5.3.2 System Specification and Design	21
2.5.3.3 Database Schema Design	21
2.5.4 Program Design and Development	37
2.5.5 Operational System Installation	30
2.6 Conclusion	30
Chapter 3. EVALUATION OF THE METHODOLOGY	
3.1 Introduction	32
3.2 System Evaluation	32
3.2.1 General Measures of Success for MIS and DSS	33
3.2.2 User Defined Success Criteria	34

3.2.3	Success Characteristics	35
3.2.4	Comparison of HEC Project with Success Factors	37
3.3	Consultant's Evaluation	39
3.3.1	Analysis and Design	39
3.3.1.1	Familiarisation	39
3.3.1.2	Requirements Analysis and Formulation	40
3.3.1.3	Database Design	41
3.3.2	Program Development	44
3.3.3	Implementation and Maintenance	45
3.4	Conclusion	47
Chapter 4. IMPROVEMENTS TO THE METHODOLOGY AND FUTURE OPTIONS		
4.1	Introduction	48
4.2	Methodology Improvements in Systems Development	48
4.2.1	Physical Requirements for Prototype	50
4.2.2	Description of Modified SDLC	50
4.3	System Development Aids	55
4.3.1	Fourth Generation Language	55
4.3.2	Software Aids	57
4.3.3	User Interface	59
4.3.4	Database Management Systems	63
4.3.4.1	Data Independence	65
4.3.4.2	Ease of Use	69
4.3.4.3	Selection of a DBMS Model	69
4.4	Summary	71
Chapter 5. CONCLUSION		
5.1	Summary	72
5.1.1	Familiarisation	72
5.1.2	Iterative Approach	72
5.1.3	Improvements to the Systems Development Life-Cycle	73
5.2	Concluding Issues	74
REFERENCES		76

APPENDIX A	FAMILIARISATION EXERCISE
APPENDIX B	RESOURCE INVENTORY
APPENDIX C	IMPLEMENTATION REPORT
APPENDIX D	EXTERNAL USERS & DISTRIBUTION BRANCH REQUIREMENTS
APPENDIX E	CRITICAL PATH ANALYSIS & ACTIVITY CHARTS
APPENDIX F	A Microcomputer-based System for Managing Land Information

LIST OF FIGURES

Figure	1. Comparison of the major MIS and LIS influences	6
	2. Shortened Systems Development Life-Cycle	15
	3. Systems Development Life-Cycle for the HEC LIS	16
	4. Consolidation Diagram	23
	5. Conceptual Schema for HEC LIS	25
	6. Basic LIS program modules	28
	7. Table of database design methodologies	41
	8. Modified Systems Development Life-Cycle	53
	9. Comparison of 3GL and 4GL for LIS development	57
	10. Distribution of effort throughout LIS development activities	59
	11. Accomodating growth in a network database	66
	12. Hypothetical schema relationships	67

INTRODUCTION

The design and implementation of a Land Information System (LIS) is a complex undertaking involving considerable resources. The justification for such a venture is twofold. Firstly, to provide more accurate, complete and timely information for use at either routine or strategic management levels. Secondly, to eliminate both, the extensive duplication of data across government organisations, together with the deficiencies of data retrieval in the existing paper filing systems.

Land Information Systems are in many respects closely related to, and have been plagued by problems and failures similar to those faced by Management Information Systems (MIS) and Decision Support Systems (DSS). There has been considerable research and empirical studies conducted in an attempt to understand the nature of these problems and formulate strategies to overcome them. In most cases the research efforts concentrate on the technical aspects of systems development and propose technically oriented solutions.

There is, however, a growing awareness that some other, less tangible, factors have a greater impact on the design, implementation and operation of Land Information Systems than merely technical considerations. This argument suggests that the interplay of people, organisations and politics is the dominant factor influencing LIS development.

Land Information Systems evolution has been influenced greatly by Data Processing professionals, therefore, it is not surprising that a technical bias has prevailed. The oponents of this theory are generally computer literate professionals, although trained in a separate discipline, and occupy administrative or management positions in land related organisations. From such positions, theirs is a global perspective, and offers an embracing view of all aspects and influences at play. Nevertheless, this argument is not widely supported. The experience and advice offered by overseas LIS 'experts' endorses the 'people-political' argument, but this advice falls on deaf ears. The LIS community, at large, remains beholden to the technocrats. Meanwhile, many LIS implementation attempts in this country continue to struggle without attaining a significant level of success.

This thesis is a case study in which the author acted as a consultant for the Hydro Electric Commission of Tasmania with the brief to undertake the design, development and installation of a LIS within the Commission's Survey section. The implementation strategy for this project was designed with the specific intention of overcoming the 'people-political' problems discussed in the section above. Importantly, therefore, the consultancy provided the opportunity, through an evaluation of the experience, to identify the critical factor influencing LIS development. Determination of the 'people-political' problem as the critical factor would justify the emphasis placed on this issue in the design of the methodology. It would also indicate a new direction for the practice of LIS implementation.

The detailed objectives of the contract were to:

- (i) devise and implement a methodology for developing a LIS for an operational environment,
- (ii) observe and analyse the typical problems encountered during systems development,
- (iii) formulate improvements to the devised implementation methodology based on its observed performance,
- (iv) make a more informed evaluation of other LIS research and empirical study.

The thesis is concerned with the design and development of an operational LIS, and has the following structure.

Chapter one examines the relationship between the development of a LIS and other Information Systems. The traditional Systems Development Life-Cycle is briefly outlined then two, more detailed, sections discuss its known deficiencies and some recent attempts to improve the methodology. The final section suggests why the traditional methodology is incapable of successfully developing an LIS, and proposes a new strategy.

Chapter two, firstly, introduces the development environment for the HEC project. It describes the organisation's structure, objectives, methods and problems. The nature of its existing data storage and retrieval systems are outlined in addition to the type of data it holds and the information it requires. Secondly, the formulation of a hybrid methodology for system development is discussed. The third, and final section of this chapter, describes each stage of the implementation methodology as it was performed in the HEC project.

The purpose of chapter three is to evaluate the methodology. This task is approached in two sections. The first of these investigates some current techniques for measuring the success of an Information System and applies these to the Hydro Electric Commission LIS. The second approach analyses the achievements at each stage of the System Development Life-Cycle and compares these with their respective objectives and expected achievements.

The purpose of chapter four, therefore, is to investigate and propose modifications to improve the implementation methodology. In addition to strategic improvements, some software considerations, such as the User Interface, Fourth Generation Languages and Database Management Systems are also the subject of discussion.

The thesis concludes with a summary of the achievements of the Hydro Electric Commission LIS project. In particular, the proposed improvements to the implementation methodology are stressed.

A more significant objective of the conclusion, however, is to emphasise three issues which, if followed, will assist those involved in LIS development toward the successful implementation LISs in Australia.

1.1 Introduction

This chapter investigates the consequences of the association between Land Information Systems (LIS) and Management Information Systems (MIS) / Decision Support Systems (DSS) in relation to the evolution of the systems development process. The traditional development methodology and some of its well known failings are outlined as are various methods introduced to overcome these problems. The chapter concludes with a discussion of some current trends in system development and proposes a scenario for improving LIS development.

The development of LIS, as a systems project, has followed the traditional process of systems analysis and design as carried out by Data Processing (DP) professionals. This association has lead LIS development through all the pitfalls that the DP profession has suffered during their efforts to build and improve the system development process. LIS pioneers had little alternative during the period of the sixties when LISs first emerged since the only technical group capable of implementing the concept were Data Processing personnel. At this time DP staff were responsible for the development of Management Information Systems (MIS) projects and computing was not well understood by lay users or management. Cronan writes ...

"many were convinced that computer programming was a nonmanageable art ... most application projects, therefore, directly depended on the programmers involved and on their understanding of the requirements of the system." [Cronan 1984]

LIS implementation arose from predominantly turnkey systems or in-house development and in both cases the DP staff were responsible for development. McLaughlin notes that end users were often dissatisfied with the systems delivered [McLaughlin 1982(a)]. This dissatisfaction was not restricted to LISs, but applied more generally to all manner of MIS [Cronan 1984, Martin 1981, DeMarco 1978, Spock 1985]. The indications were that inherent problems

in the systems development process were responsible for delivering poor results.

1.2 Traditional Approach to Systems Design

It is appropriate to briefly outline the traditional approach to system development in order to appreciate its weaknesses and the efforts made to introduce new techniques.

The Systems Development Life Cycle (SDLC) is the process through which a information management application evolves before emerging as a fully operational computerised system. Richardson [Richardson et al 1980] summarises the traditional framework of the Systems Development Life Cycle (SDLC) as follows;

- 1) requirements analysis and definition
- 2) system definition
- 3) system design
- 4) program design
- 5) detailed module design
- 6) testing and acceptance
- 7) documentation
- 8) modification and maintenance

Most theories of computer system development, to this day, usually follow these steps and although terminology may differ, Condon suggests that the basic life cycle is the same [Condon 1978].

The process begins when management becomes convinced of the need to improve the existing information processing procedures. This usually stimulates a new computer-based information system and a feasibility study is conducted, to determine if such a system can solve the existing problems, or make the necessary improvements. Based on the results of this study a decision is reached as to whether the development of a system will proceed. If so, the Requirements Definition phase is entered where, through questionnaires and interviews, analysts interpret the user's requirements to generate the system specifications. In brief, the system specification documentation, produced during the System Definition stage, contains a description of what the system will do, how long it will take and how much it will cost. This document is

also the users first and last opportunity to observe the design teams interpretation of their requirements before the system is delivered for testing.

During the subsequent stages, that is, program design, detailed module design and implementation, the physical system is developed in isolation from the users. All design and implementation functions are performed by the system design team and a programming team. The program is tested by the program team in parts and as a whole to ensure it meets the system specifications.

Upon completion, the system is delivered to the users for testing and acceptance. Provided the system 'works' the user is expected to formally accept it before any further stages are commenced. This being done, the system is documented, that is, user manuals, technical manuals, program manuals are produced and the way is now cleared for the final and on-going phase - modification and maintenance.

1.3 Deficiencies in the Traditional Approach

This approach to systems development has been employed for many years despite enormous budget blowouts, project time over runs, and widespread user dissatisfaction [Martin 1981]. Some projects fail even before completion and are abandoned, although a more common scenario is completed systems which simply fail to become a success. Such systems are unwanted legacies for organisations and become a greater liability with every modification. Keider [1981] suggests several reasons why systems projects fail completely while many writers [for example, Yourdon 1979] agree that poor design, borne from inadequate definition of user requirements, is the root cause of unsuccessful system implementations. Others [for example, Zwart 1981], suggest factors of a different nature - the interaction of human, administrative and political behaviour. For some time, however, DP authors and researches have sought technical solutions to the problem and only recently have the humanistic factors been considered for what is, in many respects, not a technical problem. For example, Yourdon observes that throughout the 1960's it was fashionable to blame systems failures on programming, as a result of which structured programming techniques were developed. Then during the 1970's the emphasis shifted toward design from which a following for structured design [DeMarco 1978] arose. By the late 1970's emphasis had had turned to user requirements, the basis of the entire system development process, as the key

to accurate design and, all other things being equal, successful implementation.

"Without an adequate statement of the user's requirements, the best design and the best code in the world won't do the job. Indeed, without the benefit of proper requirements definition, structured design and structured programming may simply help us to arrive at a systems disaster faster." [Yourdon 1979]

It is true that structured design, structured programming techniques, attention paid to system documentation, and many other factors, are all important considerations in achieving successful system development. However, it is equally true that any inconsistencies arising in the first stages of system development can only mushroom and diminish the systems' chances of success as the development process progressively builds on earlier blunders. Thus, definition and analysis of user requirements is the first line of defence against system failure and must be upheld by identifying weaknesses in the methodology and introducing techniques to combat those weaknesses. No matter how successful each of the subsequent development phases may be, if the requirements are incorrect or incomplete, the operational system will be a disaster.

A consensus of opinion [for example, Cronan & Means 1984, Sweet 1985, Spock 1985, Cullum 1985] since that time identifies user - design team communications as the principal contributing factor toward inadequate requirements definition.

"Postmortems usually show that users and data processing personnel did not share an understanding of what they were building. Not that one's outlook conflicted with the other's." [Sweet 1985]

The problem is quite subtle. The respective views of the design team and the users rarely conflict, they simply do not agree and this is a sufficient foundation for building systems which are assured to fail. It appears that either design teams can't understand or interpret what users want or, users cannot satisfactorily articulate what they want. Indeed Spock [Spock 1985] argues that both are true. It is reported [Scharer 1981] that users and design teams comprised of DP Analysts "harbour grave doubts about each other" and

recent studies indicate a severe communications problem emanating from Analyst design teams. Cronan & Means [1984] suggest communications problems in three areas, namely

- (i) composition and relationships of the DP team itself,
- (ii) relationship between upper management and DP management personnel,
- (iii) the relationship between DP personnel and the users.

The crux of the matter is inadequate, sometimes a complete absence of, cooperation and communication between various actors involved in system development. McLaughlin [1982(b)] stresses the diverse nature of organisations, institutions and actors involved in LIS and, clearly, development of such systems is impossible without some measure of harmonious participation from all these fields. The communication and cooperation problem during system design exemplifies some aspects of the humanistic problems affecting LIS development as well as highlighting the ineptitude of forcing technical solutions onto such problems.

1.4 Improving the Traditional Approach

An obvious solution is to include, in varying degrees, user and management representation on the design team itself, since DP personnel and users alike have recognised for some time that achieving an adequate definition of requirements necessitates greater user involvement. Again from the Canadian experience [McLaughlin 1982c] comes the warning that systems design driven by user needs will not become a reality by simply depositing a few questionnaires with users. However, significant progress has been achieved in Canadian LIS with an...

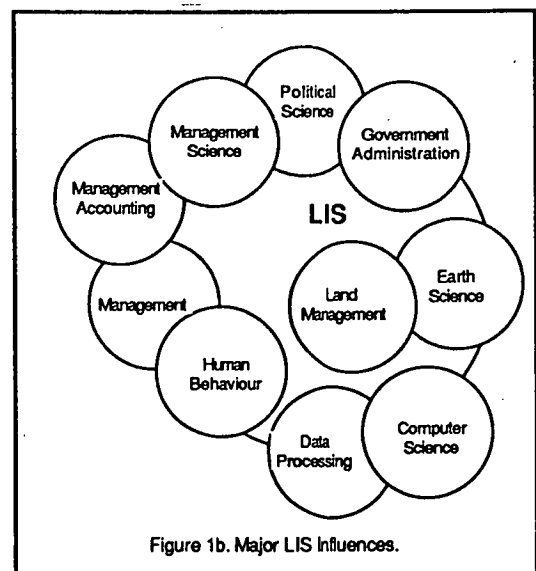
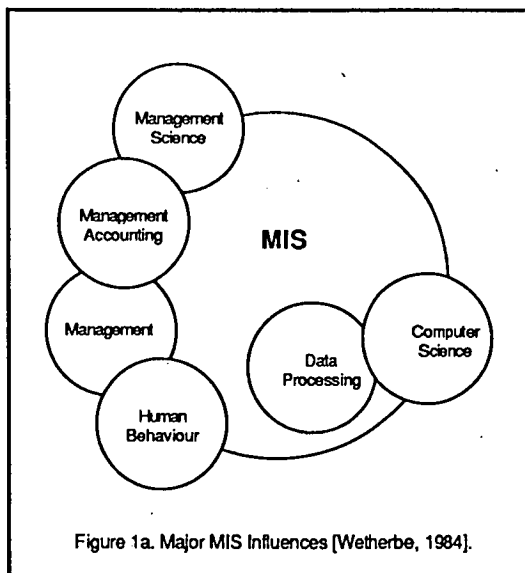
"increasing involvement of surveyors, cartographers and geographers who provide a spatial-oriented and user-oriented perspective which has often been lacking in the past." [McLaughlin 1983b]

Two recent attempts at overcoming this problem within the scope of Australian LIS development have been exercised by IBM and the West Australian Land Information System Advisory Council (LISAC). Briefly, IBM have established an Applications Transfer Methodology [Ramnath 1983, Stephens 1984] which emphasises heavy user involvement in defining user requirements and setting priorities. It involves the user at a relatively high level of the

three initial development stages and met with apparent success in the Sydney Water Board pilot project. LISAC adopted the other extreme, involving as many users as possible [LISAC 1982] in low level implementation of all stages of the system development cycle (with the exception of coding) under the guidance of a consultant.

Clearly, user involvement in the formulation of user requirements and system design is a vital ingredient in the recipe for successful implementation. There is another, more important conclusion however, in that the management and coordination of all actors in this process is the most crucial task of all. This is particularly so in the development of LISs because of the diverse nature of the data, actors and organisations involved.

For example, see figure 1(a) for MIS influences compared with figure 1(b) for LIS influences.



The MIS and DSS experience shows that user involvement has not been without its problems [Scharer 1981] and may have contributed to further fracturing of the analyst - user relationship. As users become more involved in development so they desire more control of it, thus a situation has developed where DP departments are threatened with a diminishing status.

Three initiatives from both the DP profession and user communities have recently emerged in an attempt to overcome this problem.

In many large organisations the DP department has fragmented to form a

series of Application Development departments in addition to a central DP department. These departments are responsible for developing and assisting user development of application projects while the traditional DP department is responsible for lower level system functions problems, for example, maintaining and administrating existing computer resources for the organisation. Another initiative is the distribution of DP personnel from the DP department to various user departments where they become more familiar with the function and operation of their host department and are well positioned to develop successful computer applications.

The third initiative, made possible by 4GL and personal computers, is termed User Developed Applications [Divard and Huff 1985]. In this scenario the entire project is controlled and performed by the users. The following arguments exemplify the support for user developed applications...

"users are application experts ... They are intimately familiar with the requirements, dynamics, subtleties and nuances of their work.

... no amount of insight on the part of systems analysts will ever equal the first hand understanding of the application expert.

It is ... the application experts who are best equipped to computerise their work." [Dee 1985]

The User Developed Application approach is interactive, innovative, less structured and more trial and error than the traditional SDLC. Dee [ibid] argues that structured systems development methods are related to the cost, complexity and impact of an undertaking. In practice therefore the User Developed Application senario is usually applied to personal computer based projects where these factors are less critical, although its supporting arguments are valid throughout the whole gammut of undertakings. Clearly, however, the ability of the user, the application expert, to computerise his work process diminishes rapidly with increasing technical complexity. When the undertaking becomes very complex, expensive and consequential a system development expert is required to assist the users.

Provided that the expert assists in a manner that conforms with the fundamentals of the User Developed Application strategy, the project has an excellent chance of succeeding. However, at this point the strategy, and many

other innovative attempts to improve the systems development methodology, collapses. The typical scenario is that users develop the system to a stage where the undertaking becomes too difficult and begins to flounder. Consequently management loses faith in the user's ability and quickly summons the DP or MIS department to assume control before "all is lost". The overwhelming tendency for the new regime is to apply traditional systems development techniques to revive the project. This action is probably compelled out of a conservative reaction to adopt a well known, "safe" methodology with which they feel comfortable and most confident of succeeding in their revival attempts. After all, the pressure is on the DP department to succeed where others have failed, therefore it would not seem prudent to experiment with new methodologies.

1.5 Conclusion

Problems, similar to this scenario occur when developing land information systems because they are related to managing people and the impact of technology on people, rather than technical development issues. Although there are problematic technical issues in LIS development, these do not have the far reaching consequences of problems related to the management and communication between people during such a project. Most of the LIS community, in favour of the more tangible technical issues, tend to ignore these "people" problems while only a few [for example, Humphries 1985, McLaughlin 1982b, Zwart 1981] recognise and advocate the import of the "people" problems.

To seek the best strategy for LIS development we must adopt a socio-technical approach. For too long the emphasis has been on purely technically oriented solutions for a task which poses more behavioural difficulties than it does technical. The improved techniques for analysis, design and programming are already sufficient to ensure their contribution to system success. Attention should now be focused on the formulation of a human management strategy to manage and coordinate the organisations and actors involved with LIS, to ensure their participation, cooperation and contribution throughout its development.

2.1 Introduction

From the discussion in the previous chapter is evident that the traditional approach to systems development is failing to meet the user requirements for today's modern computer applications. There is a wide consensus for this conclusion and an equally broad spectrum of techniques advocated to replace or improve the former method.

A systems developer is therefore faced with many possible approaches and the Hydro Electric Commission (HEC) system design comprised a hybrid of these. The reasons for adopting a hybrid approach rather than selecting one of the "vanilla" methods were twofold.

Firstly, and most importantly, as suggested by Zwart [1981], LIS implementation is somewhat different from MIS implementation, in particular

"Few of the management information systems ... have attempted to consolidate the collection, storage and distribution of data originating from more than one organisation let alone from a number of quasi-autonomous organisations at different levels of administration."

The HEC project is essentially contained within a single autonomous organisation however this organisation creates data of great interest to other organisations while alternatively, other organisations maintain data in which the HEC has a similar interest. It was felt that the HEC project would be, at some stage in the next five years, forced to negotiate the political and administrative inter-departmental boundaries indicated by Zwart and therefore allowance for this was built into the development process. Besides this there was a genuine desire on the part of the developer to use the HEC project as a testing ground for a LIS development strategy applicable to the broader state-wide (inter-departmental) application. Careful consideration indicated that this was also justifiable from the client's point of view (as the project was a commercial venture as well as a research and development effort), in that a strategy intended for the wider domain would also work for

a subset of that domain.

The second reason for adopting a hybrid approach arose from the special circumstances of the HEC development environment which demanded a cross section of existing techniques and notwithstanding, some innovation.

The following section describes the environment that requires this approach.

2.2 Project Environment

The HEC, a semi-government organisation, is the sole electricity generation and distribution authority in Tasmania. As its name suggests power is generated through hydro-electric schemes thus it has some rather unique land (and littoral) interests, for example; flooding easements and from a spatial perspective, long linear features such as penstocks, transmission lines as well as dense distribution networks. Land interests in general include land acquired by compulsory acquisition, purchased land, vested land, leasing (from and to third parties), easements of various types including restrictive easements, and licenses. Land use includes service depots, houses, power stations and dam sites, substations, control yards, regulator and communication sites, flooding, drainage, access and other easements, overhead, underground and submarine transmission and distribution lines, roads and offices to name but a few.

The Commission has an interest of some nature in every land parcel with power connected in Tasmania. Its land information records are therefore of immense importance, not only to the commission itself, but also other government agencies and the public at large.

Land interests, therefore, form a basis for relationships between the HEC and other organisations and the most significant of these is the transmission line easement (also called wayleave easement). The reason for its importance is twofold; firstly the time consuming legal work arising from public wayleave enquiries, testified to by the fact that this was one of the prime causes motivating management to instigate the LIS project, and secondly its relationship to the land title. The wayleave is an overriding interest on the land title, that is, not shown on the certificate but nevertheless legally effective, and as such it is the responsibility of would be purchasers to search for its existence.

The Commission is comprised of five branches each with its own infrastructure and head who is responsible to the Commissioner. All branches have interests and/or installations situated on land, or have some involvement in land matters. Each assumes autonomy and authority to make and implement independent land related decisions. In actual fact all land policy, management and correspondence is officially coordinated by the Land Administration Section (LAS) under the auspices of the Chief Surveyor. The ability of this section to establish and administer Commission land policy, however, is often compromised by other branches wishing to administer a policy of their own. The problem is exacerbated by the situation where land data collection is distributed and duplicated throughout most of the branches, each branch assuming ownership of the data it collects, even if it is not the official collection point.

It is not a coincidence that these symptoms of administrative intrigue are remarkably similar to those mentioned by Zwart [1981] in the context of LIS spanning more than one autonomous or quasi-autonomous organisation. The political interaction between branches within, an organisation, and between organisations is very similar.

HEC land administration functions carried out by the Land Administration Section (LAS) may be summarised as follows;

- (i) Manage and safeguard the Commission's land interests,
- (ii) Record the Commission's land interests,
- (iii) Carry out the Commission's land transactions,
- (iv) Answer public and Commission enquiries relating to Commission land interests.

Further, it is the role of the LAS to provide land information and related advice in the formation of Commission land policy.

In its role supporting land related enquiries the LAS receives between 30 and 60 enquiries per day from the public alone while a fluctuating number are received from within the Commission.

Like most organisations the HEC maintains a large, complex, manual filing system, archives of its land interests and all manner of related matters in order that it may respond to enquiries from within and without the organisation. This system suffers the inherent weaknesses typical of large

manual filing systems, such as, excessive storage area, inadequate cross referencing, inconvenient location, access only by personal visitation, intricate and sometimes obscure access to documents. The information service is operated on an enquiry counter basis staffed by records section personnel who are 'expert' in the use of the filing system. Other Commission personnel who are frequent users and similarly expert in operating the system (for example, LAS staff) are granted status to make independent behind the counter searches in the interests of efficiency. Nevertheless, the system is often difficult to use even for such 'experts' and obtaining data can therefore be a long, tedious and occasionally fruitless process.

The LIS project was initiated within the Survey section. A series of problems and potential problems threatening the section made management aware that an improved system for managing land information was necessary. Management within the survey section sought approval from the highest level to contract an independent consultant to design and implement such a system.

2.3 Description of the Actors

Management

As far as the project was concerned, Management in the organisation exists at two levels, namely

- (i) Operational level management: Chief Surveyor,
- (ii) High level management: Commissioner and Commission Secretary.

Users

The user group comprised representatives from many branches within the Commission. The most active and involved users were by necessity the three LAS staff, while the survey and drafting section had on average three members continually involved. Representatives from other branches included engineering management staff which fluctuated in number, up to twelve, and frequency of involvement between weekly and monthly.

System Design Team

The design team comprised all users involved in design at any given time plus the consultant.

System Development Team

The development team comprised the consultant and a part time programmer.

Project Leader / Manager

The consultant (author) filled the roles of project leader / manager.

As discussed above, a hybrid of techniques was adopted to formulate the systems development process because of the special circumstances of the implementation. In summary these were:

- Land Information Systems have additional and modified requirements for systems development.
- The project was to be designed and developed by a single consultant.
- The consultant was independent of the user organisation and not a Data Processing professional.
- The project was to be developed and installed on equipment other than the central mainframe.

2.4 Selection of a Systems Development Life Cycle

The SDLC, upon which the HEC project was partly based, contains the now well known stages, namely

- (i) Feasibility Study,
- (ii) Requirements Definition
- (iii) System Specification
- (iv) System Design
- (v) Program Design and Development
- (vi) System Test
- (vii) Implementation and Maintenance.

The GIS design model proposed by Calkins [1972, in Johnson 1981] (modified by Reed [1976, in Johnson 1981] and later by Johnson [1981]) recommends an approach which differs only in some steps within the corresponding stages. However it should be noted that Calkins model deals only with system design and not program development.

The HEC project because of its special circumstances required a slightly modified approach to the general cycle. The major circumstance, and one which continued to affect all stages, was the combination of system designer, developer, programmer and manager into the one person. Obviously, it would be beyond the capability of a single person to manage and perform each and every step to the extent that is intended, consequently some, less relevant, steps in Calkins GIS design model were either omitted or treated superficially as appropriate.

A further circumstance was the decentralisation of the project from the main decision making body of the organisation. A private consultant was hired by and, responsible to, a section head rather than a group of corporate managers. Many decisions could be taken in direct oral consultation between the consultant and the section head, with a corresponding reduction in emphasis on generating detailed reports. Thus the developer could concentrate on productive activities rather than becoming absorbed in the process of supporting the methodology [Linstone 1984]. Although an efficient avenue in terms of time and effort this course must be taken carefully to avoid certain dangers, such as omitting documentation of important decisions and their justification.

Recent DP literature proposes a modified SDLC to suit projects with limited available resources. King in his book Current Practices in Software Development [1984], recognises the need for tailoring the SDLC to suit such environments and acknowledges that implementation and management of the SDLC will depend on the people available and the type of project under consideration. He therefore proposes a shortened SDLC for "small projects" with a structure as shown in figure 2.

However, neither the SDLC nor the shortened SDLC is entirely satisfactory for LIS development of any nature and scale. Johnson [1981], in his research to improve the GIS design model comments on the disappointing contribution from MIS & DSS toward overcoming problems in GIS design, other than to find

that both fields share similar problems. This, however, is not surprising since GIS development has been based on the same principles as MIS & DSS and

Feasibility Study	System Definition	System Development	Implementation & Production
Feasibility Study	Requirements Definition	Program Design and Development	Implementation and Production
	System Specification	System Test	
	System Design		

Figure 2. A Shortened SDLC for Smaller Projects [King, 1984]

therefore will experience the same problems. Another reason is that the traditional SDLC is oriented toward event processing systems, typified by transaction processing, stock and inventory control, for example, and therefore a fundamental difference between LIS and other Information Systems is the nature of the basic data. The LIS is based on the land parcel, a dynamic but stable entity in that it is fixed in space and time. Most other Information Systems are designed around a basic unit which is dynamic and transitory, given birth at the occurrence of an event, moving from state to state at the occurrence of other events and finally passing from existence at a predetermined terminating event. It is therefore proposed that while the fundamental steps of both traditional MIS design and GIS design model are similar, the steps within these stages involve differing techniques for achieving similar ends.

A design model was finally composed by adopting philosophies from Calkins, King and Dee. The final model is shown in figure 3.

2.5 Implementation of the System Development Life Cycle

This section provides a detailed account of the HEC LIS development lifecycle.

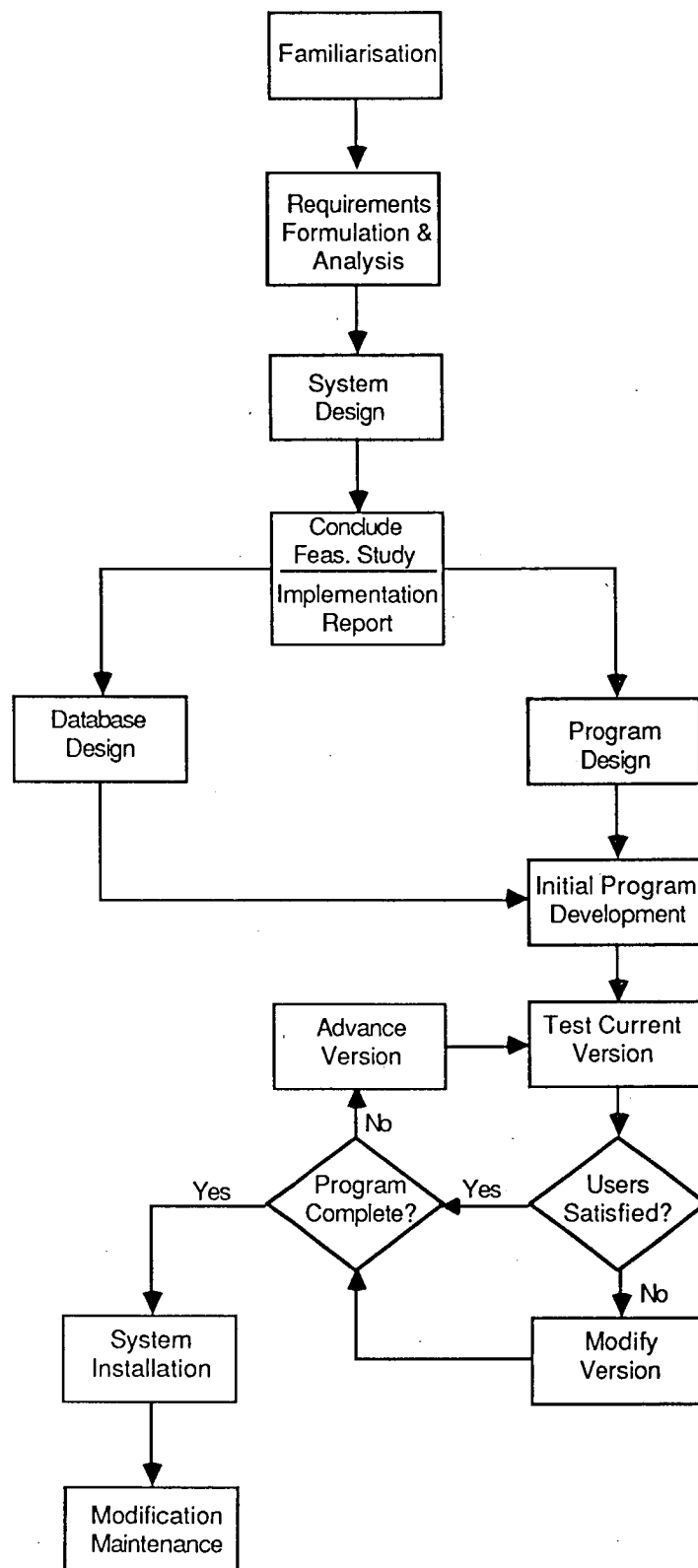


Figure 3. Systems Development Life Cycle for the HEC LIS.

2.5.1 Implementation Plan

This first step entails the formulation and documentation of an implementation strategy to define each step of the development process and create an overall context. The Implementation Plan document is a detailed expansion of the SDLC describing every step in the process from the issuing of the job brief to the final evaluation of the achievement and is as much for the developer as it is for management and users. It gives the project a sense of purpose and the direction through which the objectives will be achieved. System development personnel, management and users all need a shared vision of the project which will organise and instruct every step toward the common goal [Naishitt 1984]. The document was submitted to management for consideration and approval and as soon as this was received the system development cycle commenced.

2.5.2. Familiarisation

Traditional systems development does not include a familiarisation step at all but assumes either sufficient knowledge of the organisation and its information on the part of the design team, or that the knowledge will be gained during the Requirements Analysis and Formulation phase. Structured Analysis [DeMarco 1978, King 1984] however, recognises the necessity of familiarisation and incorporates it as an integral and continuing part of requirements analysis. Calkins' [1972], modified model incorporates familiarisation through major design stages 1 and 2, although none of the authors coin the term 'familiarisation' nor is it a separate step in their methodologies.

The object is to familiarise the designer with the operation and function of the organisation and most importantly the people who comprise it. This will take the forms of guided tours of the organisation, individual and group meetings and informal interviews with management and users where they discuss their roles, the organisations role, tools used, filing systems, enjoyable, tedious and routine aspects of their work, all of which contributes to form a general picture of what the organisation and its staff do and how they do it. Familiarisation is a vital part of the design process since it is during these informal moments that the users are perhaps the most lucid in their thinking.

most articulate in their expression of areas of dissatisfaction with current procedures, and most constructively imaginative concerning areas of improvement. This is the time the developer forms the strongest insight toward system requirements which he must later summon during system design.

Familiarisation can only be beneficial to persons already familiar with the organisations general area of application otherwise the amount of new concepts and information is too vast for a designer to process. (For example, an analyst with considerable experience in payroll systems would be overwhelmed when confronted with the new concepts and approaches involved in land management organisations). This does not suggest that such analysts could not perform the task, but rather that the ability of the analyst to share the same understanding of the problem as the user is severely diminished. Ideally, design requires people, experienced in the participating organisations general area of application who also have systems experience and analyst skills. For LIS design this requires a group that is skilled and experienced in management and administrative functions of land agencies, as well as technical aspects of system analysis design and implementation.

Familiarisation is a period of intensive interaction between analyst and user, where the analyst is primarily in a learning mode, probing for information, constructing both the macro and micro views of all aspects of the organisation.

It is most important that during this process the developer makes a written record of what he learns so that during each session he may identify, question and clarify, any inconsistencies that may develop between his and the users' views. Such interactive dialogues are essential ...

"The period over which (analyst - user) communication is turned around ... needs to be reduced from months to minutes. I quite literally mean that fifteen minutes into the first meeting ... there should be some feedback. If the task at hand requires the user to describe his current operation, then fifteen minutes into that session is not too early for the analyst to try telling the user what he has learned." [DeMarco 1979]

In addition, after each session the analyst should carefully document

everything he has learnt then immediately circulate this to the appropriate users for verification. If something is wrong or omitted it must be rectified.

At some stage during the familiarisation process in the shortened SDLC, the exact point is not clear, the developer will realise that the exercise is contributing more directly to requirements analysis and formulation. This indicates that an initial saturation has been attained and the developer now has sufficient working knowledge of the organisation to interpret and recognise facts as requirements. At this point the familiarisation documentation is evaluated and finalised and the design team moves into the system definition stage.

2.5.3 System Definition

System Definition, the second stage in King's shortened SDLC, is the process where the user requirements are formulated, analysed, translated into system specification and a general system design is evolved. Calkin's GIS design model achieves the same purpose as stages 1 and 2 of King's shortened SDLC (or stages 1, 2, 3 & 4 of the traditional SDLC) however when compared, an interesting discrepancy between the two becomes apparent. The foremost stage of the traditional SDLC is a feasibility study then followed by design, while in Calkins' model the design phase concludes with the findings and recommendations of the feasibility study. The traditional SDLC initially studies the feasibility of the project as a whole then produces a single design based on analysed requirements. Calkin's GIS model treats the entire design process as a feasibility study which generates several designs suitable for the analysed requirements, the most appropriate of which is selected as a result of the study. Calkins' treatment of the feasibility study was incorporated into the HEC design process.

2.5.3.1 Requirements Formulation and Analysis

User requirements were determined through a series of questionnaire and interview sessions with all users involved.

Three separate questionnaires covered the fields of System requirements, Information and Application Requirements. During this process, however, it became evident that the Application and Information questionnaires should be

combined.

The aims of the questionnaires were to formally establish

- objectives and functions of the participating organisation's sections
- processes by which these are achieved
- problems in these processes
- reasons for the persistence of problems
- possible solutions
- identification of data characteristics
- data deficiencies
- individuals expectations of a LIS
- identification of information used at various decision making levels
- wish list of system requirements
- acceptance criteria

In formulating and analysing requirements, Linstone's multiple design perspectives (Technical, Organisational, Personal (T-O-P) perspectives) [Linstone 1984] approach was adopted. Questionnaires were compiled by extracting questions and concepts from various references [Martin 1981, Mumford and Henshall 1979, Teorey and Fry 1982, Wenig 1982, Curtis and Jones 1982] where each contributed primarily to one particular perspective thus a cross section of questions were documented, which later would respond to analysis using Lindstone's approach.

Firstly, the System requirements questionnaire was issued to users and a timetable drawn up for interviews to commence a week later. Participants were instructed to complete the questionnaire and return it to the consultant, the day prior to their interview, in order that the consultant could prepare follow up enquiries for the interview. The interviews were conducted in private between the user and the consultant during which time each response was considered in more detail and matters arising were pursued. Each interview was tape recorded thus enabling the consultant to concentrate on interview techniques and following up users ideas and responses without being distracted by having to take notes. At the conclusion of the first interview, participants were given the second questionnaire and an appointment for the second interview.

The consultant replayed the interview dialogues soon after each session in order to document a written summary. These were distributed to the

corresponding interviewees for verification and where necessary modified to accurately reflect the users view, usually the following day.

Resource Inventory

Paralleling the Requirements Formulation process was the compilation of an inventory of resources. In the main, resources comprised of hardware and software available for the project [See Appendix B, Resource Inventory]. Also important, as indicated by Johnson, are user attitudes in the context of resources. Positive, enthusiastic user attitudes are a very rich resource indeed, while a negative or indifferent attitude is analagous to defective hardware - sometime it works and sometimes it does not but in either case the results should not be trusted. Such an attitude is a constraint rather than a resource, however it cannot be discarded, instead it must be the subject of a great deal of attention, maternally nurtured and carefully manipulated until it becomes a resource.

2.5.3.2 System Specification and Design

The stages of System Definition, considered above, yield the information necessary to lay down the specifications of a complete system. The formulation and analysis of user requirements yields 'what needs to be done' while the resource inventory yields 'what is available to do it'. The system specification and design document [See Appendix C] details possible system solutions that will satisfy these requirements given the known constraints. The document, with recommendations, was submitted to management for approval prior to contining with the subsequent SDLC stages.

2.5.3.3 Database Schema Design

Database design can be divided into three phases; conceptual, implementation and physical design. The objective of the conceptual design is to produce a schema:

- (i) representing the users' total data structure and information requirements,
- (ii) in a way which is totally independent of all physical limitations,
- (iii) is intelligible to the user.

Implementation design, is the process of transforming the schema from a conceptual information representation to a physical database structure, however the point at which this transformation becomes the third and final phase is somewhat obscure. This final objective is to produce a physical database which is organised in such a manner that optimal efficiency is achieved for the organisation's particular hardware, software operating environment and information application requirements.

2.5.3.3.1 Conceptual Design

The methodology adopted for conceptual design was based on conceptual modelling constructs discussed in Teorey and Fry [1982] namely, object representation and entity modelling. Essentially these techniques allow people to express the way they "think about", that is model, the complex structures of the real world.

The various approaches to conceptual representation of an organisations information requirements can be distilled into two major design methodologies: entity analysis and attribute synthesis. The respective approaches correspond to top-down and bottom-up design strategies.

Entity Analysis

Briefly, entity analysis (top-down) assumes that the aggregate of data across the organisation is too vast to be tackled all at once, therefore it is partitioned into various design perspectives. Teorey & Fry choose four such perspectives namely:

(i) Corporate Perspective: involves the information requirements of the organisation, as depicted by management staff, and relates to information used to measure performance, set corporate policy and high level corporate management.

(ii) Operational Perspective: involves information requirements at the organisations routine level of operation, that is, day-to-day functions.

(iii) Information Perspective: depicts the information requirements and processing necessary for decision support, ad hoc queries and long range planning.

(iv) Event Perspective: involves time and scheduling requirements for

data processing and information delivery.

Each perspective is applied across the whole project domain therefore at least two problems remain. Firstly, there remains a vast amount of data in each perspective, and secondly within and across the perspectives there are local views which have individual perceptions of the perspective. Such local views usually identify with functional areas within the organisation. In accordance with top-down methodology the domain is further reduced by considering requirements of local views within each perspective.

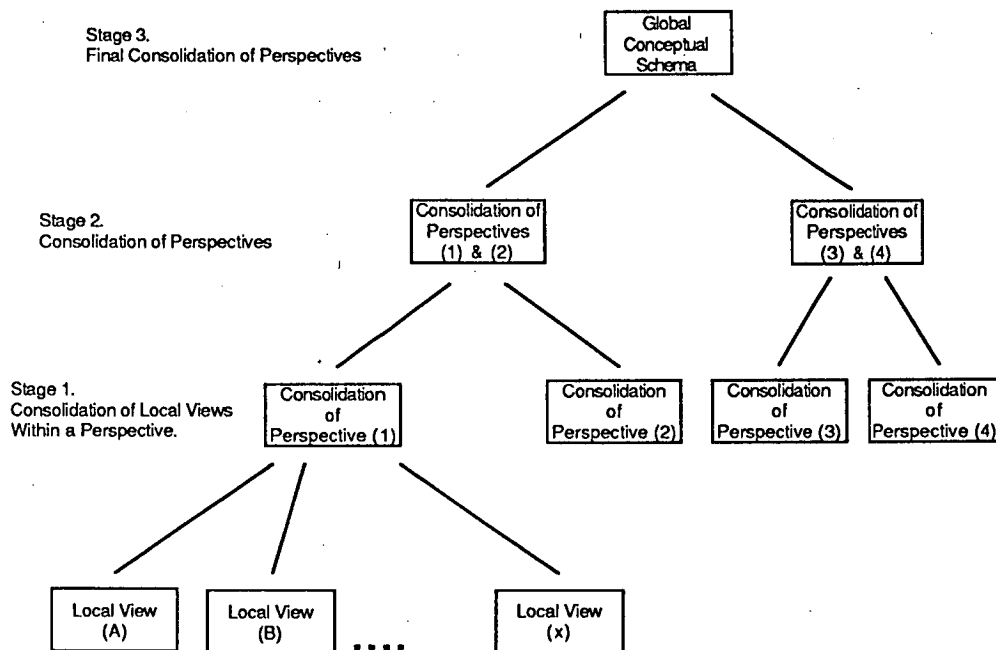


Figure 4. Consolidation of Local Views and Perspectives within the organisation to form a global Conceptual Schema.

The information available in each local view is an abstraction of the underlying data. The task of conceptual design is to apply the modelling constructs to the information in order to reveal the supporting data structure. Of course this presupposes that there is a fixed data structure underlying the information - a supposition that is not necessarily true. The data model is an artificial representation useful because neither man nor computers presently have a better way of representing the real world. "We use the model merely as a key to unlock the system" [Linstone 1984:13,14].

Entities, attributes and relationships are abstracted from the information held in each local view and represented graphically using an Entity - Relationship (E-R) diagram. This process is conducted across all perspectives within each local view. In order to achieve a global conceptual schema all views must be consolidated [figure. 4]. [See Love & Zwart 1983]

Attribute Synthesis

This method [Rund 1984] uses the data definition table formulated in the Requirements Formulation and Analysis stage and from this, constructs the high level entity, attribute relationship structure. There are four stages of attribute synthesis:

(i) Classification of data elements

Data elements from the data definition table are identified according to a set of heuristic rules as being one of two types of attribute. An identifier attribute is one that is unique across all tasks and other data elements and further may be used to access other attributes. A non-identifier attribute is one that is dependent on an identifier attribute for access and cannot exist meaningfully on its own.

(ii) Composition of Entities

The next stage involves the composition of entities from attributes. There are two types of entities, unique and non-unique. Unique entities are not dependent on other entities or attributes in any way although they must have at least one identifying attribute. Non-unique entities are dependent on other entities for existence and on other attributes for meaning.

(iii) Identification of Relationships

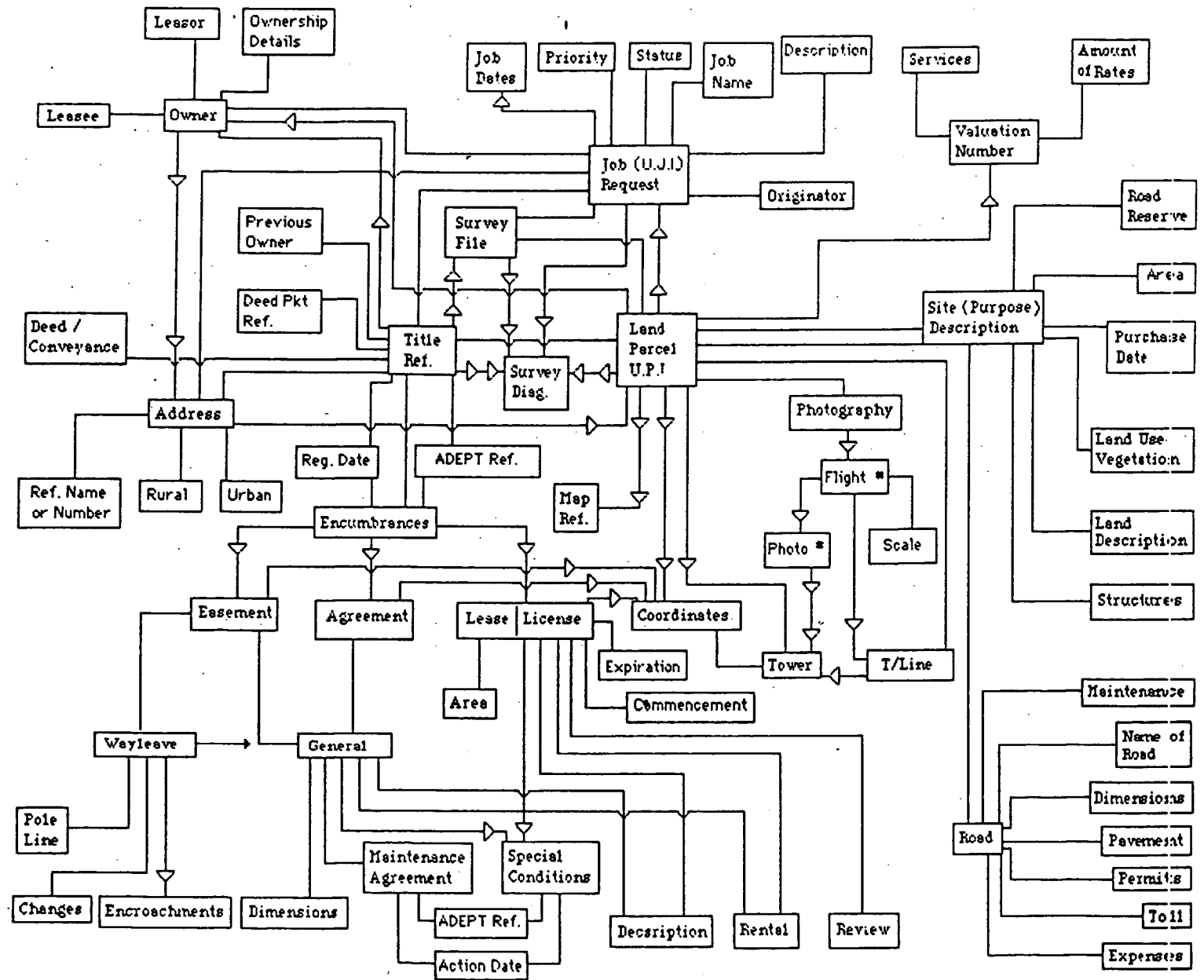
Information gleaned earlier during Requirements Formulation and Analysis is used to determine relationships between the entities. Such information may include semantic rules, organisation policy or observed interaction between attributes and/or entities.

(iv) Graphical Representation

Finally the entities, attributes and relationships are depicted graphically using the E-R diagram.

Either method may be used to attain the common goal of a global conceptual schema. It is very doubtful, however, that if both methods were applied to the same requirements data that the same result would be achieved.

Figure 5. Conceptual Schema for the Hydro Electric Commission LIS.



The former (entity analysis) approach is by admission of its authors "not scientific by any means [but] this procedure provides a design sequence utilised by some practical (expert) designers" [Teorey and Fry:82]. Clearly this is a somewhat intuitive method subject to the persuasion of the designer while, on the other hand, attribute synthesis is somewhat inflexible for a task where the output is particularly subjective.

"On each decision, the mathematical analysis only got me to the point where my intuition had to take over." [Jenson R.P in Linstone 1984:54].

Since semantics and the importance of data change from one user to the next it must be recognised that there are influences, other than those purely technical, at play in schema design.

In his book Multiple Perspectives for Decision Making, Harold Linstone suggests that material used for decision making should be considered from multiple perspectives, namely those of Technical, Organisational and Personal. Constructing a conceptual schema, in fact the entire system design and development cycle, is an especially intensive decision making process which could well benefit from such a broadly based approach.

The presence of multiple perspectives will help make all members of the design team more comfortable in their role on that committee. Traditionally emphasis has been on the Technical Perspective due to the dominant DP staff role in system development and clearly this will threaten users and management on the team since they are largely ignorant of technical issues. Similarly, dominance of either of the other perspectives will unbalance the committee and consequently de-stabilise the system design.

Database design for the HEC project essentially adopted the entity analysis methodology but also incorporated some techniques from attribute synthesis. Hybrid or multi-methodological approaches were frequently adopted during the HEC project. Reliance was rarely based entirely on any one methodology but spread over a number of complementary approaches.

"... but any extensive use of, or trust in, any methodology can only lead you astray, and you can get the help which new methods and machines offer only if your consumption remains sparing, your

methodological diet balanced and your attitude sceptical." [Kellen in Linstone 1984:15].

Adoption of some aspects of attribute synthesis seemed suitable for two reasons, firstly a list of all data elements was available, as a product of the earlier requirements formulation and analysis stage, and it seemed appropriate that this be used as design input. That is, the data dictionary provided an incomplete set of data elements to which information from the various perspectives provided rules with which to abstract entities, attributes and relationships.

Secondly, the results of synthesising attributes assisted in verifying schema development when using the alternative top-down method. Conflict, however was always reduced using information from data semantics, corporate policy or user clarification but not using heuristic rules since these are generalised technical rules and are not sensitive to context.

The conceptual schema, resulting from the above process, [figure. 5] is an Entity - Relationship (E-R) diagram which readily translates into an implementation schema for the Extended Network database.

2.5.4 Program Design and Development

Design and Development Philosophy

Structured programming techniques were employed to design and develop the program. Coding however proceeded through a series of check points, or stages at which the program was returned to the users for testing, evaluation and verification. During traditional systems development user involvement with the project ceases abruptly at this point where DP takes over. When users next see the 'system' it is completed and too late for alteration. Cronan [1984:24] however, suggests that even when user involvement is stressed, users themselves have little desire to be included in the programming stages and it is sufficient that they are only called upon for clarification and 'walkthroughs'.

The advocates of User Developed Application [for example, Dee 1985, Divard and Huff 1985] and Prototype [for example, Segal 1984, Young 1984] strategies oppose this contention arguing instead that the user is intimately

involved throughout this stage interactively designing and developing the program using DP as an "extension of the user's expertise" [Dee 1985]. The suitability of these approaches is dependent on the software environment. 4GL software is required for a strict implementation of User Developed Applications, whereas the use of 3GL high level languages is somewhat cumbersome. Although 3GL is adaptable to prototype strategies provided the longer cycle time for prototype versions is acceptable.

Design and Development Description

Simplistically the LIS program comprises three modules:

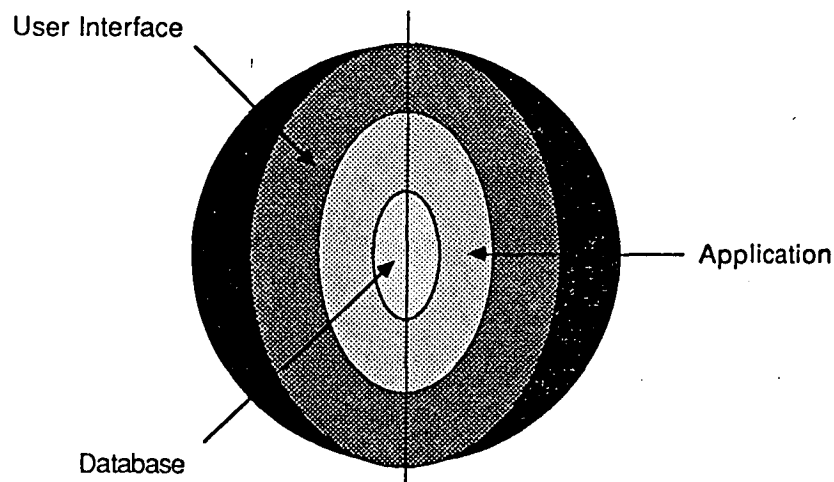


Figure 6. Basic LIS Program Modules

The user interface determines the form of interaction between the user and machine, the database interface is concerned with low level data manipulation and transfer between the program and the database, while the application management module controls all LIS operations. Of these modules only the user interface is of interest to the user since it is this part of the system with which they will have daily interaction. Therefore, maintaining user involvement throughout the program development phase requires a high priority given to the rapid development of a user interface in order that the users may

have a window of understanding through which they can view the development of the remainder of the system. The only means available to the user to understand and evaluate program development is by "hands on" experience. Thus the user interface must be developed first so that users can test the other modules as they are developed.

It was anticipated, and later verified, that users could neither envisage nor design a user interface control system. The consultant recommended and adopted a window orientated approach similar to the of the Apple MacIntosh and Lisa computers. However, once users are appropriately trained they have definite ideas with respect to presentation of their data. The consultant therefore established presentation guidelines, conducted the appropriate training and allowed users to design their own window presentations. As well as other benefits (outlined in the following chapter) this had the effect of maintaining user involvement in the project while the consultant was developing the program code.

The user interface was designed while management deliberated over the report and recommendations of the feasibility study. Once the design was complete several program utilities were constructed to enable rapid development of windows, that is, forms through which users can arrange and present data. Following this the initial code for the user interface screen management module was developed. Rapid development of this stage was essential so that the system would be at a stage where users could once again become involved. With this initial code completed it was possible to create and display a data window on the VDU and with appropriate training the users could learn to design and create their own windows.

While the users were thus engaged, the interface screen management code was undergoing continual development and as new versions were debugged, a new version of the program was made available to users. In this manner the users gained "hands on" experience of the system as it was developed and gave regular critical appraisals which enabled the developer to modify the system to their requirements while still under development.

Once the user interface was developed sufficiently to occupy users attention, development began on design and coding of the data management and database interface code, initially allowing data to be entered and updated in the system. When this was operational, in its most rudimentary state the data

retrieval code commenced in order that data, once stored, could be retrieved and displayed via the user interface. Once the new functions operated reliably the system was returned to the users for testing using actual data and the user designed windows.

This iterative, pseudo prototype, strategy was continued throughout the program development stage. After each check point the consultant returned to program development immediately implementing modifications which, arose from user assessments, and further advancing the system to the next scheduled check point. In the mean time, users retained the latest version of the software for continued evaluation. When all system specifications were implemented and evaluated, development moved to the final stage of the SDLC.

2.5.5 Operational System Installation

System installation was a relatively short process involving two activities, namely minor debugging and enhancements. Both activities were conducted by locating two workstations next to each another, one operated by a user the other by the consultant. The user operated the LIS to perform normal operations while the consultant observed. When a bug occurred or modification or enhancement was desired, the consultant edited the appropriate program on the second workstation, recompiled and made the new version available without interruption to the user.

Installation also required the development of system utilities which automated necessary functions such as; low level disk management operations, database integrity checks, database security copies and LIS program initialisation and execution.

2.6 Conclusion

This chapter has described the project environment for the HEC LIS and detailed the steps of the system development life cycle. The SDLC adopted appropriate features from a number of information system design aids. King supplied the basic framework while Calkins contributed to the system design.

with specific regard to the feasibility study. Database design techniques were adopted from Teorey and Fry, Chen, Rund and the emphasis on iterative program development was contributed by Segal, Dee, Young and Huff. Installation techniques were an innovation by the consultant.

Clearly much attention has been focused on achieving an optimum systems design and implementation methodology. It is worth noting, that in the final analysis, the SDLC will not have been followed as intended, some aspects will have received more attention than planned while others may have been omitted altogether. This is inevitable because regardless of the extent of research devoted to perfecting the design model there will always be unexpected events, special circumstances and unforeseen subtle influences that will, from time to time, divert the project from its intended course. Each implementation experience contributes to the SDLC but until that time the new considerations are unknown and ignored.

"Nothing ever becomes real until it is experienced - even a proverb is no proverb to you until your life has illustrated it." [John Keats]

Like a skilled craftsman, he works to a basic blueprint, but each production will be slightly different because of the unique circumstances presented in its case. To this extent it may even be a mandatory constituent, for system success, that the SDLC is not followed precisely since this would indicate that at least some entities peculiar to the system under design have been identified and considered. Of utmost importance however, is that the systems development methodology does not become an end rather than a means in itself.

3.1 Introduction

The evaluation of the design and implementation methodology is a difficult task since a search of the MIS and DSS literature reveals that conclusive methods for measuring success in such systems are scarce. There is an abundance of research and literature analysing why systems fail which may be indicative of the ratio of failures to successes. There is also considerable research contributing "factors which facilitate success". The HEC LIS, as an operational system, is yet in its infancy as are the users in their experience with it. It is considered that in order to conduct a reliable evaluation the system must be fully loaded with all data types and have been operational for at least six months. Although neither the system nor its users have reached this level of maturity an attempt is made to evaluate the system and its development methodology. This is approached by applying the "measures of success" detailed in MIS and DSS research literature and examining the acceptance criteria detailed by users during the Requirements Definition phase of system design. This will be dealt with in the first section of the chapter. An alternative approach is also discussed which attempts to perform the evaluation by investigating the existence of the well known factors causing failure and those believed to facilitate success during the implementation experience.

It is anticipated that the above section will not produce conclusive results thus the second section of this chapter involves an objective evaluation from the consultant in order that additional indicators may be revealed.

3.2 System Evaluation

Two system evaluation techniques are briefly discussed, both involve an analysis of the system with respect to certain measurement criteria. The first set of criteria are generated externally [Sanders 1984:33] and are a general set of evaluation criteria for DSS. The second are user defined criteria

originating from the questionnaire and interview sessions during the Requirements Definition phase [Appendix C, Chapt. 5].

3.2.1 General Measures of Success for MIS and DSS

Sanders [1984:29], suggests two important criteria for measuring success of MIS and DSS implementations:

- (i) productivity
- (ii) user welfare.

Productivity is measured by,

- (a) profitability (the net gain between cost and benefit),
- (b) relevance of the system to the major problems of the organisation,
- (c) improvement in quality of decisions.

User welfare is measured by user satisfaction - a very difficult factor to measure. One such possible measure is the degree to which the operational system is used. In this case widespread use infers great user satisfaction and system success, whereas infrequent use or abandonment infers the contrary. Several writers [for example, Schewe 1976: 557-590, Ginzberg 1981] denounce use as a reliable measure although Lucas [1978:27-41] disagrees provided it is voluntary. If use is accepted as a measure then the HEC system would rate very favourably since the Land Administration Section (LAS) are enthusiastic and frequent users. However, it is also conceivable that there is a novelty period after which system use will diminish.

Considering the productivity factors (a) - (c) it is regarded that (a) profitability, per se, is not a valid criterion for a government organisation. Indeed the measure becomes less useful, whether applied to systems in government or business, the more emphasis shifts toward decision oriented information systems rather than transaction processing systems. A technical approach such as cost / benefit analysis is suitable for the latter while benefits of the former decision oriented systems are not qualifiable and not suitable for such an approach.

An evaluation of (c), quality of decisions, is premature at this stage of the HEC system operation. The importance of this measure should be emphasised, however, since there is a broad base of underlying factors which come into play in considering this measure, thus making it a good judge of the system. For example, from a technical perspective, the quality of decisions is

dependent not only on the data the user is permitted to select to support the decision, but equally importantly, the integrity of that data. A social perspective, on the other hand, indicates the degree of confidence users have in the system to deliver information in a satisfactory period of time, with acceptable presentation and ease of accessibility.

The relevance of the system to the major problems of the organisation, factor (b), has similarly not been analysed and evaluated over a reasonable trial period. However, from experience thus far the system effectively addresses itself to the organisation's major problems of (i) cross referencing information between organisational divisions, (ii) timely delivery of information and most importantly, (iii) simple access to information and simple system operation.

Users have verified that land information which previously was extremely difficult or impossible to obtain is now easily obtainable using a familiar parcel identifier. Tests performed by LAS users indicate that retrieval time for certain land holding categories is approximately equal to the quickest look up time for the simplest case land parcel when trained staff use the manual system. Parcel retrieval is many times faster than when untrained staff use the system or when the level of difficulty increases, for example, the location is not accurately known. According to one user, speed and accessibility to land information is sufficient to answer a telephone enquiry 'on the spot' without noticeable delay. The third criterion, simple access and operation, is a conclusive factor indicating system success. This is evidenced by the users ability to operate the system without the requirement for manuals or user training sessions. Clearly the user interface and the prototype approach to system development facilitated this measure of success. Throughout program development users participated by frequent testing and evaluation of prototype versions, therefore, when the system became operational they were already familiar with it. New users, only remotely involved during system design and development, have also remarked upon the ease of its use, thus further testifying to this indicator.

3.2.2 User Defined Success Criteria

One section of the questionnaire and interview sessions during the Requirements Definition stage involved the construction of a list of user generated acceptance, or success, criteria. The primary purpose of this was to

formulate specifications which could be used to define the limits of the project. That is, satisfying the criteria would also indicate satisfactory completion of the contract, thus releasing the consultant. From an analysis perspective, however, evaluation of the criteria represents a measure for system success.

Unfortunately, as stated earlier, it is too early in the operational life of the system to give meaningful answers to most of the criteria. They are, however, noteworthy because of their relationship to other criteria in MIS and DSS research. Several authors [Henshall 1979: 119, Segall 1984: 143, Oppermann 1984: 157] dealing with system success characteristics agree that success must be both technical and social. Social requirements are perhaps the most difficult to satisfy since they involve satisfying the interests of many and varied user groups throughout the organisation while also fulfilling the personal requirements of each system user. They are not only difficult to satisfy, but also, difficult to define and are therefore frequently neglected, or merely paid 'lip service' during design and implementation. The requirements definition stage of this project revealed the importance of social requirements as a contributing factor to system success since each of the six acceptance (success) criteria were social rather than technical in nature. (See Appendix C: Chapt 5).

In this regard the design phases were proven to be thorough and successful in revealing and defining important criteria that are commonly neglected.

3.2.3 Success Characteristics

The discussion above, although indicating some areas of success, does not conclusively endorse widespread system success. Further indicators may be yielded by analysing the design, implementation and operational periods of the SDLC in search for characteristics common to successful systems.

McCosh [1984:20] in his study of twelve DSS implementations, attempted to identify the system characteristics which indicate that a system is likely to be successful. He concluded that there are three categories of systems:

- (A) those that fail without becoming operational,
- (B) those that fail after becoming operational,
- (C) those that succeed.

These will be referred to as categories A, B and C successes. He also observed significant differences in the characteristics of each system during implementation. Category A was easily distinguished from the other categories as it displayed very few success characteristics while categories B and C were much more difficult to separate since they had many common characteristics. McCosh concludes that three characteristics must be present during implementation:

(i) A System Champion

The system must have an enthusiastic main user who strives to have the system accepted and operational. He must have the resources to achieve these ends, that is, authority to allocate resources or, alternatively, considerable influence with others who do have the authority. Further, the champion must be patient and committed, able to suffer setbacks and recover without losing faith in the project.

(ii) Top Management Support

DSS and MIS literature has reported for many years the necessity of top management support for the project. [Keider 1981, Condon 1978]

(iii) User Involvement

User involvement is now recognised as being a vital component in all phases of systems development and a major factor contributing to system success.

However, McCosh's observations show that these characteristics do not, necessarily, ensure operational success. He argues that once a system enters the operational phase a new set of factors begin to influence the system. The most significant of these are social factors. McCosh cites an example where the superior of the main user supported a system during design and implementation, although he was privately concerned with regard to his own ability to control the technology. However, once the system became operational his anxiety was realised and in a defensive attitude he withdrew the support of his section from the system. Inevitably, without political support, the project collapsed.

The advantages of the prototype development method (discussed later) are intended to minimise the risks of this type of problem.

Other studies [Shrivastava 1984:136] concur with the observation that new

factors are introduced and threaten the system life support once it becomes operational. Shrivastava notes in one example that the redistribution of information caused by the operational system led to new power alignments and in the ensuing re-organisation those who lost power tended to be obstructionist and non-cooperative [Shrivastava:138]. It is interesting to note that the potential for both scenarios exists in the HEC operational LIS environment and present possible risks to the system. However, as suggested, an approach was adopted that seeks not to offend the superiors of the main users by imposing demands on them which they have not been trained to meet, rather they are given the necessary time and training to learn.

3.2.4 Comparison of HEC project with Success Factors

System Champion

The HEC project displayed category B characteristics throughout design and development. Initially the consultant performed the role of champion, however, a user champion eventually arose from the LAS. Surprisingly, an actor external to the organisation (such as the consultant) could perform the role of champion, however, the example of an internal actor has a great deal more effect on other actors within the organisation.

Management Support

Management support existed through the section head to the Commission secretariat. This was absolutely essential, especially in light of the fact that the normal DP department authority channel had been circumvented. Visually however, management support played a low key role - support was there when required and if justified although, from a user perspective there was little evidence of this.

User Involvement

User involvement, as previously emphasised, was a key factor in the design and development phases. Apart from requirements analysis and system design, where current literature concurs that user participation is essential, the methodology also included users during program development vis à vis the prototype technique. There are additional benefits derived from prototyping that have not yet been discussed. In particular, benefits which contribute directly to the social success factors. Segall [1984:146], identifies eight advantages of prototype methods, three of which relate to requirements definition and are not dealt with here while most of those remaining relate to

benefits accruing during program development and system installation. Segall notes the following benefits,

(i) No Surprises - since users have been involved in the program development, testing and evaluating each successive version, the final system will not hold any unexpected 'features' or omissions [Ross and Schoman 1979].

(ii) Experiment - ability to experiment with users ideas that would not normally be investigated because of the uncertainties and expenses surrounding them.

(iii) User Confidence - involvement in program development builds users' confidence that the system will be implemented to their satisfaction. At each iteration users observe and evaluate the effect of their requested modifications thus more intimately involving them and eliciting a greater commitment and loyalty toward the system from them. This factor in particular, contributes directly to the psychological success of the system. In other words, the system is a success in the eyes of users if it does what they expect it to do. As user confidence grows the system may gain some additional champions.

(iv) Aids Training - when the system becomes operational, users will require little additional training since they are already familiar with it. The fact that six months after the HEC LIS was commissioned, users had not required reference to operating manuals, testifies to this.

These advantages contribute to the social success of the system as they concentrate on maximising the relationship between user and system. Users will identify with some intimacy, to a system which reflects a concept or feature which each user recognises as his own. Such a relationship may lead to the user becoming a champion whence he will staunchly support the system against all scepticism and criticism from its adversaries. Not all users develop a loyalty so deep, however, the prototype methodology does instil some of these characteristics and certainly enhances the system success factor.

It was anticipated that analysing the project within the framework of the success measures and characteristics found in current research, would not determine conclusively whether or not the system was a success. It is concluded that the system is a category (B) success, that is, it exhibited successful design and development characteristics, but the success measure for installation cannot be assessed accurately at this time.

The remaining section of the chapter evaluates the development and installation from the consultant's perspective, comparing the expected results of methodologies with the results achieved in order that the project might be analysed in the widest possible context.

3.3 Consultant's Objective Evaluation

The methodology, although flexible, innovative and a hybrid of the most appropriate characteristics from other standard SDLC methodologies was still not without deficiencies. Evaluation is facilitated by considering the project as three stages, namely 1) Analysis and System design, 2) Program development and 3) Implementation and maintenance.

3.3.1 Analysis and System Design

3.3.1.1 Familiarisation

The methodology adopted was very effective in this stage. From the outset, during Familiarisation, it succeeded in quickly educating the consultant in the functions, objectives and methods of the organisation. This period also performed the task of 'breaking the ice' between the consultant and the users since not all users were enthusiastic, some indifferent and sceptical, some secretly opposed and others were not even aware that the project was under consideration. This was a particularly important achievement since it was considered that the subsequent system development step, questionnaire and interview sessions for requirements analysis and formulation, could not proceed until the rapport between consultant and users was sufficiently friendly and cooperative [Shrivastava 1985].

During the course of familiarisation the iterative nature of the methodology, feeding back to the users for verification the knowledge that the consultant was gaining [DeMarco 1978], proved a most valuable feature. On many occasions, through such feedback, users corrected the consultant on subtle issues where their mutual understanding might otherwise have been disparate while at other times the revelation of additional information would result. The use of diagrams to represent work processes and information flows

was also extremely useful since a diagram overcomes all terminology barriers and represents simply what would otherwise complicate the dialogue.

3.3.1.2 Requirements Analysis and Formulation

The questionnaire and interview sessions were also very effective. In particular, this step achieved the roles of;

- formalising and adding to information gained from the familiarisation process
- identifying corporate objectives, functions and methods,
- identifying weaknesses in the existing system, reasons why they had not been previously overcome and discussing possible solutions,
- identifying and analysing data requirements,
- establishing a 'wish list' and a required list of system features,
- construction of a data dictionary defining a preliminary list of data items to be held in the system.

Again the feed back of interview summaries to the users was of great benefit in verifying and clarifying information.

In summary, this first stage of the project up to and including the production of the feasibility study and report of the implementation proposal, appeared to have proceeded very well. A clear picture was portrayed of the organisation, its objectives and the means for their fulfilment. Sixteen key problems in the organisation were identified and many more of a subsidiary nature. Furthermore, the reasons for the existence and persistence of these problems were revealed. Users contributed many possible solutions to the problems in addition to a wealth of other information relating to system requirements, expectations and desired benefits. A near exhaustive data definition matrix was established, and both a 'wish-list' and a list of essential system applications were formulated.

In retrospect, however, it is evident that not all system requirements were ultimately identified. The failure was not widespread, in fact, it was confined only to the identification of applications (enquiry reporting). It is almost inevitable that one or more requirements will be omitted and not realised until the system has been operational for some time. It is also more likely that such omissions will not occur in the data management component, since

this has the most well known requirements, but instead will lie in the applications enquiry component. It was observed during Requirements Definition that users experienced most difficulty in anticipating enquiries and applications of the nature which occur at regular frequencies but have long periods. Chapter 4 discusses improvements to this portion of the system development methodology.

3.3.1.3 Database Design

A recent study [Chilson and Kudlac 1983] investigates and reports on familiarity and use of various database design methodologies in a broad spectrum of organisations. The report concluded that database processing is becoming an integral part of many organisations' data processing and decision making functions; a conclusion with which many other authors concur [Curtis and Jones 1984:163]. The use of methodologies for formally designing these databases, however, is far from prevalent.

Logical Database Design Methodology	Familiar With	Using
Bubenko	14.8	1.6
Chen	41.0	9.8
Curtice and Jones	23.0	6.6
Data Base Design Aid	65.6	23.0
Gerritsen	4.9	1.6
IDD as a Design Tool	19.7	3.3
Kahn	8.2	3.3
Navanthe and Schkolnick	8.2	1.6
Sheppard	6.6	1.6
Smith and Smith	6.6	0.0
Teorey and Fry	9.8	0.0
Other	31.1	31.1

Figure 7. Familiarity and Usage of Database Design Methodologies [Chilson and Kudlac, 1983].

Note that figures do not total 100 percent due to the fact that some organisations indicated familiarity with and/or use of one or more design methodologies.

The study indicates [figure 7] that, of the current database design methodologies the manual approaches are least used although respondents were familiar with some of them. Date [1983], reported to the 10th Australian Computer Conference that few logical database design aids were available

and, in any case, these were treated with some scepticism by database designers since the aids require as input all the data a designer would require to perform the task manually. Date argued that the result would be more reliable if the latter case were adopted.

Teorey & Fry note in their book Design of Database Structures that much of their work is built upon earlier research by authors whose methodologies appear in figure 7, for example, Chen, Sheppard, Smith and Smith, Navenhe and Schkolnick, [Teorey and Fry 1982:61,66,76]. The methodologies of these authors are in most respects very similar and had they been combined in figure 7 the conclusion may have been that nearly all respondents were familiar with this type of data modelling. Even so, the number of respondents actually using it would still remain below 20%. Despite the considerable research effort invested in manual design methods they continue to be largely ignored. This is unfortunate since the conceptual schema is intended to be used and understood by users. It provides the context for which each user group can observe their specific data and the inter-relationships with other users' data. Furthermore an understanding of the schema is absolutely vital for meaningful application of database query languages and report generators.

Conceptual schema modelling for the HEC LIS, using data modelling constructs and combination of Entity Formulation and Analysis [Teorey & Fry, 1982] and Attribute Synthesis approaches [Rund, 1982], resulted in a very satisfactory and rapid development of a highly accurate conceptual design. User scrutiny of the conceptual schema identified only two misrepresentations within the 102 relationships and 70 entities on the diagram.

Evaluation of this process highlights three issues, namely

(1) Translation of User Requirements data to Database design

The final accuracy of the conceptual schema and the ease with which it was constructed, is primarily due to the communication level between database design and a previous SDLC stage - Requirements Formulation and Analysis. The form of the data collected at the earlier stage was such that it readily translated into database design specifications. For example, the existence of a complete data definition matrix formed an immediate basis for a data dictionary and a starting point from which to identify entities, attributes and relationships. When this was combined with other questionnaire and

interview data, such as corporate and routine objectives, functions and methods the Entity - Relationship (E-R) diagram was smoothly established. A more critical analysis of questionnaire and interview data was necessary to establish the E-R diagram where data was common to more than one section of the organisation, that is, existing in multiple local views.

(ii) Transformation from Conceptual to Implementation schema

This transformation is noteworthy since it highlights compatibility between the Extended Network database model (for example, MDBS III) and the conceptual schema depicted by the E-R diagram. The transformations required to implement the conceptual schema, for such a database, are minimal since both models support identical data structures. In fact, the process may require only the utilisation of special database features such as indexes, hashing, ordering within relationships, and database security measures.

The transformation of the conceptual schema necessary for other databases, whether hierarchical, CODASYL network or relational, will require application of special rules and the resulting DBMS processible schema will have a different representation of the information. For example, these database models require cumbersome structures to represent many to many relationships, and the relational model necessitates the introduction of attribute redundancy to form any relationship between entities. Users performing ad-hoc queries with a query language will then be required to understand the transformation from the conceptual to physical schema in order to correctly execute enquiries. This is an additional burden which could confuse users and result in queries yielding spurious information.

(iii) Freezing completed schema design

It is recommended by database design experts [for example, Landry et al 1985:35, Teorey and Fry 1982:138] that the conceptual schema is frozen once completed. The conceptual schema is the information base which remains unaltered unless additional facts are presented which show the schema to be in error or to contain an omission. This is a particularly important rule to be remembered when considering consolidating entities from a one to one relationships to a single entity. It is often favourable from a database point

of view but may alter the information base. In the event that a proposed schema modification alters the conceptual schema semantics, the design must then return to the requirements definition stage to redetermine the information for this portion of the schema [Teorey & Fry, 1982]. The physical schema should not be necessarily frozen since modifications may be necessary to take advantage of special features of the hardware or software.

3.3.2 Program Development

The program development stage proceeded using the iterative prototype methodology described in an earlier section of this chapter. Adoption of this methodology, although effective, could have been more effective had certain other factors not been prohibitive. For example, the period between successive iterations should, ideally, be almost instantaneous so that users could immediately observe the effects of their requested modifications. Obviously this is not realistic with present day software development tools otherwise an entire project could be completed within a short period with the user and developer sitting together at the VDU the whole time [Cullum 1985]. In the future, something of this nature may be possible (or more likely system development will take a completely different form) however for the time being we must be satisfied with simply minimising this period. Fourth generation software tools allow major modifications to be completed during the same day as the request is made, but the third generation development language (MS-Pascal) adopted for the HEC project usually required several days. The period between software versions containing modifications and significant advances usually required at least one week. Although Pascal is considered a High Level Language, the development environment it creates is still a generation behind the integrated 4GL databases currently available. (Reasons for the adoption of Pascal are given in Love [1985:524].)

The 8088 16/8 bit hardware also contributed to this problem since long delays were experienced while programs compiled. This factor was anticipated therefore it was intended that a more powerful co-processor be introduced to alleviate the problem. Unfortunately, contrary to certain advertising, availability of this hardware did not eventuate for the development machine.

A similar argument exists for the development of the user interface.

Users designed their screens and windows with the use of special graphical pro-formas. Screen design could be performed interactively but required editing a text file of coordinates and attributes. After editing the file was analysed, loaded into a screen database and finally an image file created after which it could be displayed and manipulated as a window on the VDU. This process was reasonably quick for the consultant, some 2 -3 minutes to effect several minor changes, but not truly interactive and thus not an effective tool for users. To be such, it must give users the ability to interactively construct and edit the window image on the graphics screen while the corresponding manipulation of the coordinate and attribute file is performed invisibly in the background.

The most significant problem identified during program development was attributable to the particular Pascal implementation employed. A technical constraint is enforced which restricts a program in its use of memory allocation: all data, stack and heap usage must not exceed 64K bytes. The constraint arises because of the 64K address space of the Intel 8088, 8086, 80286 family of central processor units (CPU). A more sophisticated implementation of the compiler would provide automatic mechanisms for addressing memory beyond the 64K limit. At least one recently released Pascal compiler provides such an enhancement, however, at the time program development for the HEC project commenced this compiler was not available.

On two occasions the 64K limitation temporarily terminated program development, however, the program was successfully revised to make more efficient use of memory. Nevertheless, at the point of maximum memory usage during program execution there remains a mere 5K of free memory. This could have an impact if program enhancements are required which necessitate significant portions of code being appended to the program. Although, further program revisions, with the intention of decreasing memory requirements, may alleviate this.

3.3.3 Installation and Maintenance

The parallel system installation methodology (chapter 2) was very effective, enabling the consultant to debug and enhance the system in less than three weeks without interrupting users. However, during this period a flaw in the project implementation plan was revealed.

Sophisticated databases such as MDBS III maintain consistency checks on all areas of the database to ensure data integrity. An abnormal program termination is one situation that may leave the database in an inconsistent condition. This is not a problem provided the database has a transaction logging and roll back facility, or a recent backup is available, to restore or replace the corrupted file. Unfortunately the HEC had decided to forego the expense of this MDBS utility, furthermore, the streaming tape backup hardware proved to be unreliable. Therefore on the initial occasions during installation when an abnormal program termination occurred the database could be restored only by replacing the current file with the most recent backup. On more than one occasion this process failed due to problems encountered with the streaming tape device. Of course, this problem would not have been a consideration had the recovery and transaction logging utility been purchased or even had the back-up device been operating reliably.

Another problem, however, became apparent when restoring the database using a recent backup. It is possible for certain area(s) of the database to be inconsistent while not affecting others. For example, say the main database area is consistent as are all other areas in which a user enters during an update session. In this situation the database will operate normally even though, say, one area which the user does not enter, is inconsistent. At the end of the day this database is backed-up according to standard procedures. Now should this back-up be required at a later date to restore a corrupted operational database there will be an inconsistent area which, when a user attempts to enter, will deny him access thus rendering the database useless.

In order to overcome this problem a number of database utilities were constructed to monitor database status before and after each user session. This prevented an inconsistent operational database to be backed up and consequently overwrite a 'clean' backup copy.

The HEC LIS project, like all systems development, was subjected to many problems and irritations during its development lifecycle, however, without doubt the most serious of these was the database recovery problem. The demoralising effect this had on both staff and management was serious indeed and, should it have persisted, would most likely have terminated the project. Each failure represented the loss of several weeks work to data entry staff and management became under great pressure from higher authority to give reassurance of project success.

Unfortunately the recovery problem was identified after the event and only then did the subsequent experience lead to the development of utilities which prevented the situation from recurring. It is not an exaggeration to state that these utilities saved the project, alternatively, had they been developed in anticipation of operational problems then the situation would not have arisen and the system spared of much scepticism and embarrassment.

3.4 Conclusion

Like the evaluation of the first section of this chapter, this assessment indicates the system to be, at least, a category (B) success in terms of the design and implementation methodology. Some failings of the installation process were identified, although they have been overcome. Limitations caused by the Pascal compiler were shown to have some effect on program development and could yet require further attention.

Whereas these problems are technical by nature and may be overcome by technological advances, a more significant problem was the failure of the requirements definition stage to identify all enquiry and application requirements. This represents a failure in the system development methodology and may be surmounted only by re-evaluating the SDLC and formulating a new development philosophy.

The introduction of this chapter indicated the intention to evaluate the design, implementation and operation of the system in order to judge its success or failure. It was considered that the system had not been operational for a sufficient period to pronounce operational success although the indicators suggest that this may be the case. It is suggested, however, that the achievement of this chapter has been, not to pronounce success or failure on this case study, but to investigate and apply various means of evaluating the success of a land information system from its adopted design and implementation strategy to its performance as an operational system.

Chapter 4. IMPROVEMENTS TO THE METHODOLOGY AND FUTURE OPTIONS.

4.1 Introduction

The HEC LIS was a carefully and thoroughly planned project with significant attention given to improving those areas of system design and development where traditional practices have often failed. The evaluation in the previous chapter concluded that the methodology was successful, although certain problems and deficiencies were identified. The purpose of this chapter is to investigate and propose alternatives for an improved methodology based on the HEC LIS experience. In addition to these improvements other options such as software tools and comparative database structures are discussed.

4.2 Methodology Improvements in Systems Development

The improved model seeks to redress the three deficiencies of the previous SDLC, namely :

- (i) uncertainty of defining all user requirements during requirements definition,
- (ii) delay in successive iterations of program development and inadequate degree of real time interaction during window design,
- (iii) inadequate database protection.

Considering these in reverse order, point (3), database protection, is easily accommodated in program development by introducing a final step in this stage to provide for the development of appropriate tools. The second point, delays in program iterations, can be overcome by adopting a development language more conducive to the speed requirements of the prototype environment. Increased real time user - machine interaction, when designing window presentations, can be achieved by either purchasing or developing a more sophisticated window design tool. However the proposed solution to the first, and most significant, of the deficiencies necessitates substantial changes to the SDLC.

Identification of user requirements is the spectre of all systems projects; it is virtually impossible to be confident that all such

requirements have been identified during this stage. A SDLC that employs a prototype approach to program development, will be less susceptible to failure due to omission of requirements. Such omissions may well be identified eventually during prototype testing and be subsequently accommodated in the prototype. However, it is expedient to identify requirements at the appropriate stage, before design or development. Accommodating requirements that have been overlooked, without consideration to the overall design, may introduce unforeseen problems and could dramatically affect time and cost estimates proposed in the feasibility study or implementation report.

The proposed model enhancements seek to increase the confidence of users and developer alike, in that all such requirements are identified and that any omissions are merely trivial enhancements easily accommodated during program development.

It has been shown already that requirements are overlooked and systems frequently become category (A) failures (fail before becoming operational) because of communications breakdown between system designer and users. In particular, this is a result of:

- (i) designers and users have a mutual lack of understanding of each other's vocabulary,
- (ii) users who have no conception of what a computerised information system might do, have difficulty in articulating their requirements,
- (iii) no quality measurement exists for good requirements definition.

Recent studies of the application of a prototype approach to systems development show that this approach can be used at any stage throughout the SDLC. For example, the prototype may be;

- (i) used as a pilot, that is, an intermediate step in a larger development process,
- (ii) used as an initial prototype in a succeeding prototype process,
- (iii) used in conjunction with traditional development process to either, aid requirements definition, or design and implement a system.

Some empirical studies have been performed to analyse the use of a prototype approach during requirements definition and the following advantages were

reported [Segal 1984];

- (i) users can experience a system before having to articulate their needs,
- (ii) a real system provides a better communications medium,
- (iii) a tangible system is easier to evaluate than a paper model.

It can be seen that a prototype system used in the early stages of the SDLC will overcome the previously observed communication problems ((a)-(c)) contributing to system failure. It is therefore proposed that the LIS SDLC be enhanced by embracing prototype methodology (ii) or (iii) above . The SDLC would then be modified as shown in figure 8.

4.2.1 Physical Requirements for Prototype

To implement this approach the designer will require a generic Information System prototype, that is, a user interface and skeleton generic Data Management Component (DMC) (see Love 1985:529) that can be quickly adapted to reflect data items owned by the client organisation.

The generic system would be best implemented using a Fourth Generation Language (4GL) relational database, interactive forms generator and query language. A demonstration user interface exhibiting the major features of the proposed system interface is easily developed. The relational database should be established as a single table representing the primary unit (land parcel) with all related data attached as attributes of the relation. Such a representation, called the "universal relation", is currently the subject of research [see for example, Brady, 1985] although it has not been implemented at an operational level. The intention here is merely to simplify prototype programming rather than attempt to establish a universal relation. In fact a detailed knowledge of relational theory and considerable data manipulations are necessary to construct this relation. It is not simply a matter of defining one table then attaching all elements to it.

4.2.2 Description of Modified SDLC Stages

4.2.2.1 Familiarisation

Again the first step in the SDLC, Familiarisation proceeds as before,

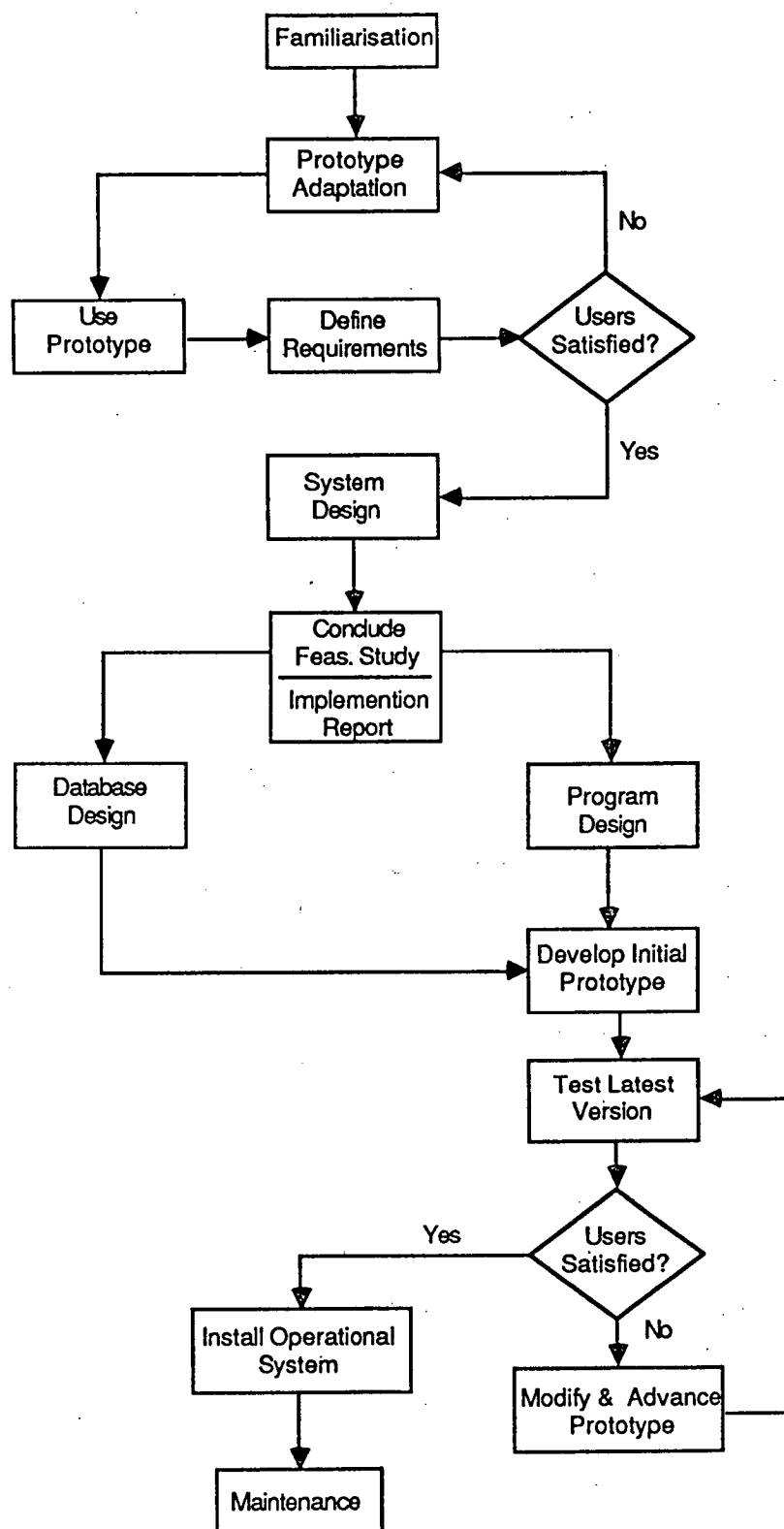


Figure 8. Modified Systems Development Life Cycle for the HEC LIS.

although it is performed with more emphasis on the organisation's data characteristics, since these will be required in a preliminary form earlier in the SDLC (for the prototype).

When the Familiarisation stage reports are completed and gain user acceptance, with regard to accuracy and completeness, the second SDLC stage, prototype adaptation commences.

4.2.2.2 Prototype Adaptation

This involves the adaptation of the generic Information System prototype to the current organisation's basic data requirements. It entails creating appropriate forms, defining attributes (data items) for the database table and linking the appropriate attribute names into the generic Data Management Component (DMC) program.

This is a relatively simple task and should be completed in one or two days at the outside. Time is essential at this stage since the users involved in the Familiarisation stage will have had sufficient exposure to the project to become interested and enthusiastic. Such positive user emotions are very important to the SDLC and must be encouraged. Any delays or sustained loss of contact between developer and users (or management) will allow them to "cool off". In order to avoid this possibility it may be necessary to overlap the final days of familiarisation with prototype adaptation.

The developer must be continually aware that the purpose of the prototype is a demonstration system to give users a "feel" for a computerised Information System, what it does, and how one interacts with it. It is not intended to be a full development system.

4.2.2.3 Requirements Analysis and Formulation

This step proceeds much as before with question and interview sessions, although with the additional aid of the prototype system. The aid has a twofold use.

Firstly to give each user a working demonstration of an Information System facsimile prior to the distribution of the questionnaires. Obviously the purpose of this is to provide users with a context within which to approach answering the questionnaires.

Secondly, to make frequent reference to the system during the follow-up

interview sessions in order that ideas may be clarified and most importantly, to extend the user's imagination when determining enquiries and applications. For example, a user developer dialogue might be as follows;

DEVELOPER: (querying a questionnaire response regarding a desired application)
"When you say you want to be able to find shack sites by their unique shack number, do you envisage something like this..."

(they turn to the VDU)

"You command the system to find a land parcel..."

(a pull down blind or pop up menu appears and the developer selects "Find Parcel" from the available options by pointing with a mouse)

"Type the shack number in the appropriate data window as the search criteria..."

(They wait a few seconds then the windows are filled with data relating to the desired shack.)

"Now the system has found your shack and displays the corresponding data. How's that ?"

USER:

"Well, actually, what I had in mind, was to display all shacks, like that, but only if their shack number is the same or greater than the one I first type in. Oh, and they should be displayed in ascending order according to the shack number."

This example, which is a great deal more realistic than most people might imagine, shows the sort of information a developer rarely reveals in an ordinary interview situation. Instead this discovery usually occurs after sign off, during installation, when it becomes a maintenance matter - hence the frequently quoted 80% [Martin 1981] project cost associated with maintenance. Clearly the revelation of this requirements data is delayed because it is very difficult to extend the user's imagination sufficiently to ascertain the necessary detail unless he visualises the actual process in question. A user who has had little, or no, experience with computer systems will have great difficulty in visualising such a process.

Once the need for a modification is verified it should be effected immediately, however it is recognised that this may not be feasible. For example, the modification may require more than a few minutes to effect in which case the developer must decide whether it is worthwhile making it at

all. For it is pointless spending more than five minutes attempting to implement new code while the interviewee waits, "twiddling his thumbs", as much as it is fruitless to code a concept that will provide no further requirements data. The sole purpose of the prototype at this stage is to present and test concepts for the user in order to assist them in formulating and articulating their requirements. Segal [1984] warns of the dangers of prototypes and indicates that if it is not to become the operational system then the correct balance of detail, providing sufficient detail to adequately model the final system but without exerting too much effort, is difficult to achieve.

4.2.2.4 Feasibility study and Implementation Report.

This step is not modified other than to suggest that all users, rather than management alone, should participate in the evaluation of these reports. Users hold a deeper insight into the operational procedures of the organisation than do management and are therefore in a stronger position to evaluate conclusions and proposals in the reports dealing with these matters. Management, on the other hand, have a more complete perception of the global (corporate) perspective and are therefore suited to evaluating project recommendations based on the corporate implications.

4.2.2.5 Program Development

Modifications to the methodology, necessary here, are minor and simply involve the addition of a step to develop database monitoring tools. Ideally it is intended that more sophisticated development tools for both the user interface and main program will be deployed, such as 4GL, graphics and user interface toolkit.

This completes the description of the modified system development methodology. The following section (4.3) discusses new and improved systems development aids which could be used in LIS implementation.

4.3 System Development Aids

4.3.1 Fourth Generation Languages

The Fourth Generation Language versus Third Generation Language debate has raged for three or four years now without either side increasing its popular support [see for example, Simpson 1983, Steele 1983, Scott 1985, Garnett 1985, Hinds 1985, McNaney 1985, Tay 1983]. DP personnel generally favour the traditional languages while some application developers and most end user groups support 4GL. It is not intended to discuss 4GL here by way of entering this debate, rather to investigate the conditions of use and possible benefits gained from the use of 4GL for LIS development.

4GL products can yield significant advantages to the LIS System Development Life Cycle (SDLC), as discussed in the previous section of this chapter, by providing a prototype tool to assist in determining user requirements. In particular, a microcomputer based 4GL provides relatively powerful and easy to learn language, simple relational database implementation, forms generation and query language. At this point in time microcomputer 4GLs do not have the power and flexibility to form serious contenders for complex systems development [Coleman, 1984] although as prototype, or mockup, systems they are ideal.

Such 4GLs must be enhanced and contain the following characteristics before being considered a suitable tool for LIS development:

- (i) a complete user interface environment: not merely forms generation ability, including high level access to menu display functions, forms manipulation and user dialogue,
- (ii) database: relational or, preferably, post relational with a complete set of features such as, security checks, data integrity checks, roll back / transaction logging, multi-user support, database status feedback, data dictionary.
- (iii) high level query language and report generator,
- (iv) high level program language, with
 - (a) interface to Third Generation Language (3GL) code libraries,
 - (b) use of system graphics kernel and interface to graphics primitives.

Such a development environment would represent a considerable improvement over the currently available 3GL tools and offers three significant advantages:

- (i) it is conducive to prototype methodology throughout systems development. Furthermore, the option for the initial prototype, developed during requirements definition, to become the final operational system may be exercised (indeed all prototype options [Segall 1984] are available),
- (ii) time reduction in some SDLC stages, especially program development,
- (iii) easier enhancement and modification of operational system which reduces the maintenance component of the SDLC.

Analysis of two recent, similar database implementations with which the author has been involved, one using 4GL and the other 3GL, indicates a 3:1 increase in productivity over the entire SDLC when the 4GL is used. A further benchmark is provided in a comparison between two microcomputer based LIS, the first system implemented using an early 4GL [Love 1982], the second is the subject of this thesis and uses 3GL. Here the ratio is 6:1 in favour of 4GL. However, although the complexity of the problems were similar in both cases, the complexity and success factor of the final system was significantly greater in the latter case. Therefore it is suggested that ratio would be reduced somewhat.

It is interesting to note that productivity ratios quoted in recent literature are often in the order of 20:1. Three factors contribute to this discrepancy. Firstly, the 4GL used by Love was a very early version of this new software generation, being particularly inflexible and unreliable, thus not greatly enhancing productivity. Secondly, a 20:1 ratio would not be uncommon where a 4GL is being compared with COBOL, as is usually the case, rather than, say, Pascal. Thirdly, it is suggested that the productivity gains available from 4GL increase as a function of the complexity of the task. Conversely, it is also suggested that at a certain complexity the gain diminishes rapidly to a point where 4GL is inadequate whereas 3GL continues to address the problem. For example it has been suggested that 4GL is currently ideal for prototype LIS development but inadequate for the operational system. On the other hand 3GL is appropriate for the final system although it is not suitable for the less complex prototype (See figure 9).

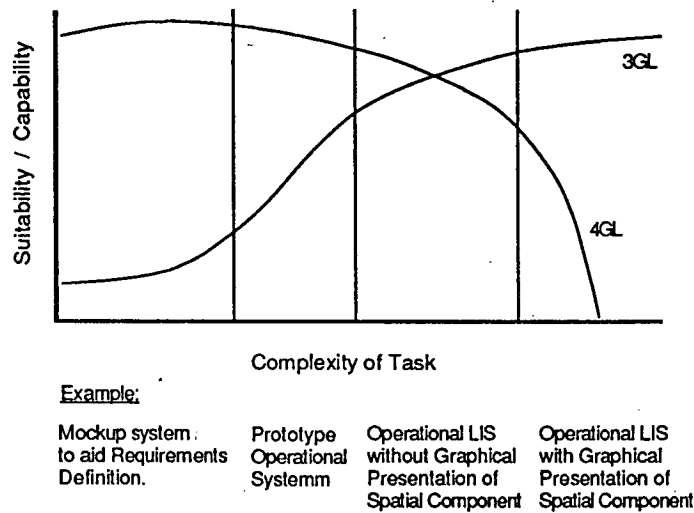


Figure 9. Comparison of 3GL and 4GL for LIS development.

4.3.2 Software Aids

A 4GL of the type discussed above, although not currently available, will certainly be developed, and probably before mid 1987. Unfortunately, while 4GL database, program and query languages are the centre of much development and market attention, the graphics capabilities of these products are usually limited to simple business applications such as pie charts, bar graphs, etc. Presently only 3GL offer the flexible, integrated development environment interfacing the database, powerful program language and extensive graphics capabilities necessary for LIS. The use of 3GL would seem to eliminate the possibility of the prototype approach to systems development advocated in the former section of this chapter. However, given the use of certain software aids this is not necessarily the case.

At its highest level of abstraction, the LIS comprises four components the User Interface, Database, Graphics and the Application. Traditionally Information Systems were constructed as very large programs and programmers were forced to develop "in-house" file systems for data storage systems in addition to coding the Data Management Component (DMC) and standard enquiries. Program languages prevented programmers from modularising the code and large complex, programs resulted where the various application functions were interwoven. However, 3GL high level languages and software tools currently,

available allow recognition of Information System components and the introduction of modularity into systems programs. Courtaz [1985] acknowledges this trend with respect to the user interface...

"There is an emerging paradigm for interactive systems architecture advocating the strict separation of the functionality of a system from the user interface."

The advantages of modularising the system are twofold, firstly, a desirable degree of independence is introduced between program modules and secondly, following from this, is the ability to develop standard generic software tools to replace various segments of in house code thus, saving a large amount of programming effort. For example, whereas programmers traditionally developed "in-house" file systems for data management we now have the generic database. The database is physically independent from the main program and is interfaced through high level function calls in the host language. The graphics component also has been modularised and interfaces in a similar manner, for example Tektronix Plot 10 graphics library and Hewlett Packard DGL (device independent graphics library). Thus, with the use of a fast compiler and appropriate software toolkits, a 3GL prototype environment becomes a possibility. However the availability of a module or toolkit for the user interface is unfortunately lacking.

The development of a sophisticated user interface represents a significant portion of code design and development. The graph (figure 10(a)) of the distribution of effort during the HEC LIS project indicates that this activity consumed 55% of effort over the entire project and 66% of effort (figure 10(b)) during program development. The availability and use of a user interface toolkit would reduce the program development stage by 27% and the entire project by 22%. Thus the developers would be free to concentrate on the application (solution to a problem) and spend less time on the details of the user interface [Bass, 1985]. Perhaps MS-Windows (to be released November 1985) will fulfil the user interface toolkit requirements under MS-DOS [Lemmons 1983], and later XENIX, operating systems but for larger systems there are no such products available.

Distribution of Effort Throughout LIS Activities

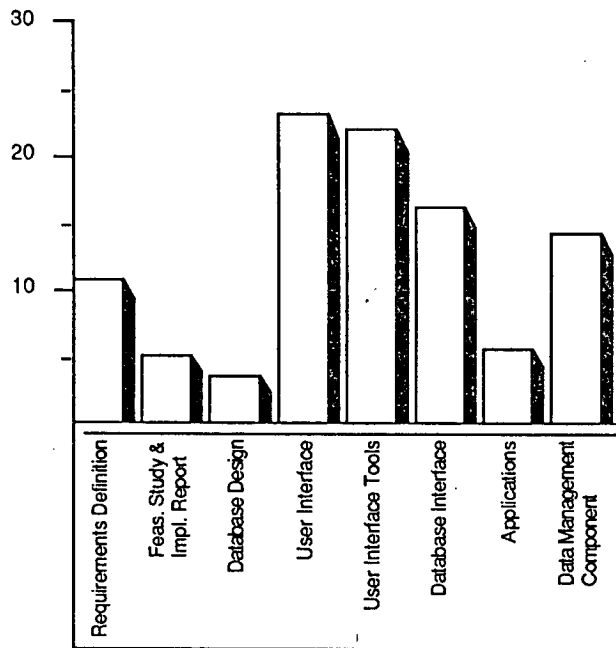


Figure 10(a).

Distribution of Effort Throughout Program Coding

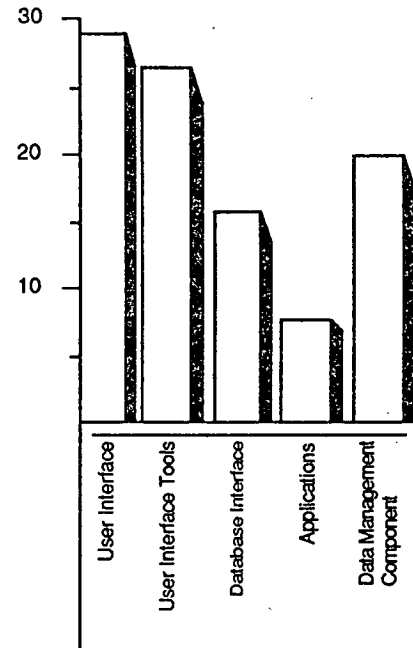


Figure 10(b).

4.3.3 User Interface

Human - Computer interaction has been the subject of considerable research since, at least, the early 1970s when the Smalltalk language was pioneered at the Xerox Palo Alto Research Centre. It later gave rise to the windowing and icon operating environment of the Xerox Star. The greatest advance, in pragmatic user interface philosophy, came with Apple's Lisa which offered its own innovations [Tesler, 1985] such as pull down menus, one button mouse and clipborad (as a means of integrating independent applications) as well as borrowing past innovations such as pop-up menus, overlapping windows from Smalltalk, and icons from the XEROX Star.

4.3.3.1 DP Attitudes

In contrast to its research attention, the User interface, until very recently has received little attention from the DP or MIS departments. Traditionally the primary concern of Information System developers has been for technical efficiency and functionality resulting from limited storage and memory [Yestingsmeir 1984]. This was usually sufficient since users of these systems were generally computer operators familiar with the intricacies of programs with crude user interfaces. The end users of modern Information Systems however are of a different breed, often administrative and management personnel with little, if any, computer experience. The demands and expectations of the user interface are, therefore, much greater.

Commercial Attitudes

The computer market too reflects little evidence, and even less support, of sophisticated and productive user interface software. This is understandable in the case of the Xerox Star workstation which, although offering a superb user work environment, demands a relatively exorbitant cost. The similarly superb Apple Lisa was introduced at an acceptable cost with a full suite of business software and yet the commercial disappointment of this machine nearly caused the demise of its creator. The hardware expansion necessary to run VisiCorp's VisiOn operating environment and associated software [Barney, 1983], actually did cause the demise of its creator.

The evidence tends to implicate the question "Are sophisticated operating environments really necessary?" [O'Connor 1983:14-15]. A recent study conducted in California [Sullivan, 1985] comprising a corporate MIS panel, indicated that operating environments were not the solution to corporate needs, indeed, the panel echoed the response of Duane Bell (a partner in the Peat Marwick San Jose office) that "nothing on the horizon will answer our [software] needs". The HEC LIS experience elicited a similar response from management who did not perceive themselves as end users. These people regarded time invested in the user interface as yielding a poor return. On the other hand

management for whom the system would represent a personal tool were very interested in the development of the user interface. A corollary emerging from this seems to be that the further one is removed from the operational aspects of the system the more emphasis shifts toward what the system does rather than how a user makes the system do it.

Therefore in the process of developing a LIS certain people, probably management who will not have direct contact with the system, will contest proposals to develop sophisticated user interface software. These attitudes should be anticipated, the individuals identified and educated with respect to the importance of the user interface to system users, before they have the opportunity to influence the disposition of other management.

There is, perhaps, another factor influencing the "community's" view toward operating environments which relates to the demystification of computing. The personal computer operator [Dee, 1985] experiences total control (and responsibility) over the computing environment; he boots the machine, mounts the disks, tends the printer, backs up files, maintains disk integrity and performs disk directory maintenance. In the traditional data processing environment technical specialists are required for each task. Thus for a time the DP department experienced a disturbing loss of control over the organisations computing facilities. Ditlea [1984], however, reports that within a short time the DP department had returned control to within its own hallowed walls. The generation of innovative operating environments currently exemplified by Apple's MacIntosh present a new threat to the DP department. This environment simply does not require the traditional level of technical expertise to safely and efficiently operate and administer the system. It quite adequately looks after itself, thus it diminishes the mystique conjured by the computer jargon used by computer experts and displayed on VDUs.

It is unlikely, therefore, that either managers whom have not had "hands on" contact with computers, or DP personnel with hidden motives and reputations to preserve, will pledge their support for operating environments that bring "computing to the people".

Many analysts of the computer industry, especially those with an

interest in micro computers, argue strongly in favour of the desperate need for improved user interfaces. A recent article [Patkin 1984] conducted a study which attempted to determine some characteristics of the "average" user in order to ascertain their probable interface requirements. Three characteristics of those identified were;

- Fifty perent of potential users are below average in mental capacity.
- The user's short term memory can retain approximately only seven items for no more than two or three minutes, even less if other messages come to the brain in the interim.
- Users generally exhibit limited patience with computers and possess the habit of "trying things out" without first studying the manual.

From these few features alone it is obvious that the human-computer interface will require specific attention.

Saja [1985], provides some formality to the argument by introducing a theoretical framework for human - computer interaction. He proposes two models, the user's cognitive model and the designer's conceptual models of the system. The user's cognitive model is formed during his interaction with the system and corresponds to what he believes are its functions, capabilities and limitations. The designer's conceptual model represents his expression and intention for the system's functions, capabilities and limitations. The user interface is the link between the user's cognitive model and the designer's conceptual model and its function is to present the system to the user, in such a manner, as to guide his cognitive model development to where it coincides with the designer's conceptual model. Only at this point does the user experience harmony in the interaction. In practice this means that the user understands where the weaknesses of the system are and has learnt ways and means to avoid these, while still satisfying his processing requirements. Furthermore he experiences full control over the system, has a "mental picture" of the command structure and can readily access and execute any system function via the fastest channels.

An invalid cognitive model, however, will result in user dissatisfaction, lack of confidence and makes inefficient use of both human and computer resources. Clearly, harmonising the user's cognitive

and designer's conceptual models results in improved user productivity and, indeed, is the sole purpose of the user interface [Love, 1983]. The point where the cost of computer time became less than that of human time was past many years ago and the gap is steadily widening [Nijssen 1983, Date 1984]. In fact, computer processing capability per unit time is increasing, while human processing capability remains relatively static. Other than concluding that humans are more expensive to employ than computers or robots and, that humans are becoming increasingly dependent upon computers to maintain and disseminate information throughout our society, the following points also emerge:

- increasing computer processing power does not necessitate increased human efficiency,
- improving the interaction between humans and computers will increase human efficiency in the use of computers,
- improving both the interaction and the application software will increase human efficiency in their tasks [Saja 1985:36, Bass 1985],

Clearly, the improvement of the user's operating environment is an important step in the overall process for improving user productivity. Its importance also lies in its influence on the category (C) (operational system) success or otherwise of a LIS. The empathy arising from the correspondence between the cognitive and conceptual models will determine a user's inclination to make voluntary use of the system. There seems to be little doubt in the minds of researchers, that the user interface influences systems success and user productivity. Certainly there is no doubt in the minds of users [Strehelo, 1984] who have experienced improved operating environments, that these systems represent a significant advance in end user computing.

4.3.4 Database Management Systems

According to Bill James (senior product manager for relational database systems at DEC's software engineering facility at Nashua, New Hampshire) there is no longer any doubt that applications involving complex data relationships must be implemented using either Relational or CODASYL Network. The alternative, hierarchical model, conceptually

holds to a very out-dated data structure which is incapable of effectively modelling complex information requirements [see Love 1983]. The debate between advocates of the network and relational databases still continues [Stamen 1985]. Essentially, the argument in favour of the CODASYL network concerns the following points, namely:

- (i) Data retrieval performance which is derived from its mechanisms enabling the database to be "fine tuned" to any application.
- (ii) Well understood concepts of its implementation and operation and thus a greater availability of CODASYL database administrators and programmers.

The relational database argument is supported by the following points,

- (i) Relational systems have built in optimisers able to decide on the fastest way to join tables. An ad-hoc query optimised in such a manner is most likely yield a better performance than the same query performed by a CODASYL database if it were not originally tuned for that query.
- (ii) Transitory data structures - Restructuring the database will be necessary because of the nature of the data. Relational implementations offer this ability to varying degrees. This capability also lends itself to an iterative approach to database design which is more encouraging to user participation.
- (iii) Simplicity of data structure - the simple table structure offers an easy understanding of the data model.

It is apparent that the CODASYL network is a very rigid structure requiring formal database design techniques and is applicable to very stable data environments. The Relational model appears more flexible, apt to interactive, interactive user developed applications and ideally suited to unstable environments where current and future information requirements are difficult to specify. The contentions regarding the CODASYL model are true, however, some of those relating to the relational model need qualification.

4.3.4.1 Data Independence

As described in chapter 2 the conceptual schema is a conceptual representation of reality with respect to the domain of the database application. The physical database is derived as a function (mapping) of the conceptual schema where the function introduces specific variables pertaining to the particular database model and system hardware requirements. Over time the nature of the application will change - requiring new enquiries, applications and data. To reflect these changes in the database the conceptual schema must be updated and the modifications mapped to the physical schema. There are two general classes of change that can occur in the conceptual schema: growth and restructuring [Date 1981].

The concept of growth is simple and falls in two categories:

- (i) Expansion of an existing entity (record or table) to include additional data (data items or data elements)
- (ii) Inclusion of a new entity (record or table).

Growth

Modifications necessitated by growth are usually simple to accommodate. The relational model incorporates a command with capabilities to redefine the existing definition of a table in order to add, remove or change the definition of an element.

The network model usually accommodates growth in the form of a number of reserved records attached to the hub entity, or reserved data items attached to certain records. These records then take on the definition of the growth data [figure 11].

The effectiveness of this method is limited by the positioning of the record in the schema, in other words, the relationships it is able to undertake. It therefore requires some accurate precognition on the part of the database designer, with respect to probable future information requirements. Alternatively, a much less trivial approach is to form a new database by recompiling a modified schema and uploading data from the original database.

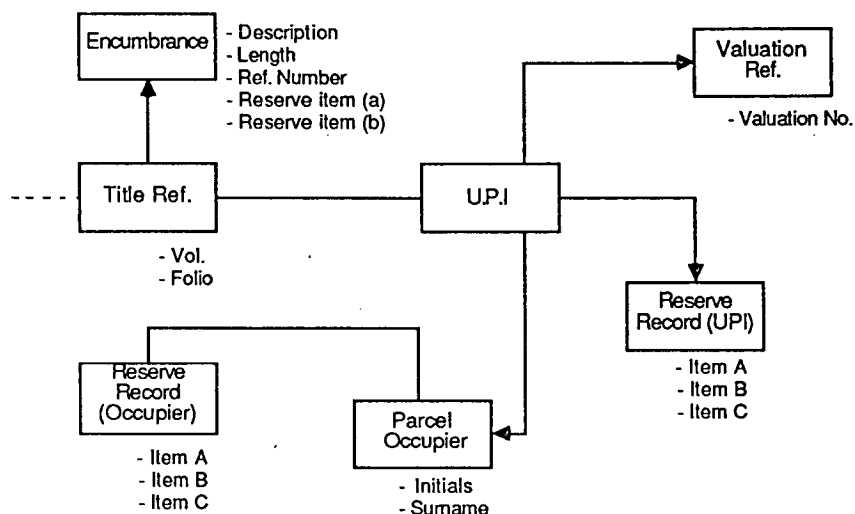


Figure 11. Reserved records and reserved data items are a simple but limited technique allowing for database growth in a network DBMS.

Restructuring

Restructuring is a considerably more complicated task necessitated by significant changes in the relationships between entities. This may be brought about by either a change in data semantics (conceptual change) or a change in the physical operating environment (physical change). An example of the latter may be simply outgrowing the capacity of a disk drive which would then require relocation of a data file to another device. The following example describes the nature and consequence of a logical change.

Say, valuation and legal parcels are currently non-identical entities in a conceptual schema. Generally the valuation parcel is equivalent to the legal parcel but frequently, especially with regard to commercial property, the valuation parcel represents a conglomerate of legal parcels. Very infrequently the situation can arise where the legal parcel is a conglomerate of valuation parcels (for example, residue land in a developed subdivision where a single title remains with the developer but the local authority values and assesses the residue as individual lots). This is a many to many relationship represented thus...



Figure 12 (b). Many to Many relationship

Say, legislation were introduced to the effect that all private property assessments were to be based on the legal parcel (title) regardless of common ownership in commercial zones or residue subdivisional land. Then the relationship would become one to one represented thus...

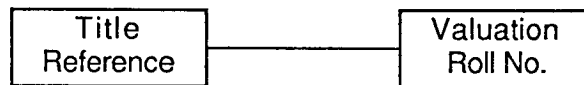


Figure 12 (a). One to One relationship

and modification to the conceptual schema would be required. Although the relationship has reduced its scope, the physical schema and application programs would not demand modification since the latter relationship is a subset of the former. However the representation is obviously not strictly correct. Had the hypothetical legislation expanded the scope of the relationship, for example, a one to one becomes a one to many or many to many relationship, then both conceptual and physical schemas and the application programs would demand modification since information in the new representation could not be accessed. The new relationship is a superset of the original.

The network model reconciles restructuring by creating a new database with the desired structure and uploading this with the original data and, as Date [1981] comments, is a "generally troublesome business".

The relational model accommodates minor restructuring using the same techniques as those used for database schema growth, however large scale restructuring can be as "troublesome" for relational models as it is for the network. There are two specific dangers associated with the restructuring capability of relational models, namely

- (i) In the process of adding or removing data items, tables of

the new structure may no longer be in an optimal normal form. Date notes that normalisation theory is not a panacea for relational database design nor it is mandatory that a database should be in 4th normal form or 5th normal form (it is mandatory that it is in at least 1st normal form). However it is desirable to be in 3rd normal form in order that relations exhibit "good" properties of redundancy and structure.

(ii) The ability to readily modify the schema may lead to sloppy initial database design. Clearly the knowledge that design errors can be relatively easily removed is an invitation to ad-hoc database design. Whilst an iterative approach to database design is encouraged in the earlier chapters it is also recognised that an undisciplined, ad-hoc approach will result in a highly patched system where it is impossible to reconstruct the context of database design decisions.

(iii) No amount of restructuring can correct certain errors. There are certain conceptual design errors that, when translated into a physical database, cannot be corrected (not without a program developed specifically to administer that modification).

For example, say a table includes a field capable of holding up to thirty alphabetic characters, and labelled "owner" being the registered proprietor of a freehold title. And during data capture, data is entered in the following manner; firstly the initials of the owner then the surname. Clearly the error in this design is that if the exact initials of a surname are not known to a user then great difficulty will be experienced in attempting to locate that occurrence. The problem could be partially overcome by simply entering data in the reverse order, surname then initials so that the user can browse through all occurrences until the surname with the desired initials is located. A better approach would be to create separate fields for both initials and surname. A user could then search using the maximum of known data available without resorting to the use of wild characters, which result in the retrieval of considerable surplus information, to represent the unknown data. This requires not only the addition of a new field but also the transfer of intrinsic data embedded in one field to another.

It could be argued that, in this example, data independence was neglected because of poor design and therefore is not an issue. However, this would simply endorse the danger expressed in point (ii) above. Design errors occur - "to err is human" - and no degree of data

independence will rescue some such situations.

4.3.4.2 Ease of Use

The second contention regarding the relational database requiring qualification relates to its ease of use. Date [1981], indicates that the tabular structure of the relational database is the key to its simplicity and [James 1985] concurs in that this characteristic enables less experienced database users to create a database application more quickly than a team of CODASYL database administrators and programmers.

On the other hand Date also reminds his readers that database design "can be an extremely complex task" and ...

"It (still) requires skilled people to make a database implementation hum. More importantly, it requires these people to know just what the DB software is doing to make the application efficient."
[Keyes 1984]

There is some consensus then that "...a DBMS is not tool for an end user" [Dewhurst 1984], not, at least, so far as the design and implementation of complex databases is concerned. Since "experts" are required for large database implementations, the simplicity and ease of use may be of limited value to LIS development and perhaps even a potential hazard if it is concluded that database applications can be left in inexperienced hands.

4.3.4.3 Selection of a DBMS Model

The choice of DBMS for complex Information System development is not, as James proposes, limited to either relational or network. Several experts, even strong relational advocates such as Date, acknowledge the potential of other approaches...

"Indeed there is widespread agreement that neither (Relational nor Network) approach is adequate in itself for the task, but rather that some extended formalism, such as Chen's "entity-relationship" model is needed." [Date 1981]

The entity relationship model, often termed post-relational (referring to its evolution from relational database theory) differs from the traditional relational model in its use of relationships. A relationship between entities (tables) is formed by association of attributes participating in the host entities. The relationship may be formed by an expression involving the attributes rather than the usual "equi-join" operation where the join condition is specified by equivalence of attributes in the associated tables. The relationship (join condition) is named and defined in a data dictionary whereafter it is referred to by that name (similar to a network set relationship). The post-relational model has several advantages, namely

- (i) support for many to many relationships,
- (ii) rich constructs to capture the data semantics of an application [Chiang 1983],
- (iii) high level of abstraction,
- (iv) natural translation from conceptual to implementation schema,
- (v) mechanisms for achieving performance,
- (vi) convenient database navigation derived from named associations.

Another "extended formalism" is the extended Network (for example MDBS III used by the HEC LIS project) which, as its name suggests, is founded on the CODASYL Network. Despite its network based architecture this approach exhibits identical advantages to those stated above for the post-relational model as well as the additional abilities to support;

- (i) recursive relationships,
- (ii) sorting relationships without physically creating index files.

On the other hand it continues to suffer from the network's cumbersome management of database growth.

It is suggested that the future LIS developments, at least at a research level, should lie not with either relational or network but a model derived from the extended formalisms. The principal justification for this is the ability of these models to directly represent the conceptual schema. The post-relational model would be the most favourable choice because of its greater data independence.

4.4 Summary

The evaluation of the system development methodology in the previous chapter has indicated some weaknesses arising from the Requirements Definition stage. It has been suggested that users experience difficulty in articulating and later verifying their requirements because of the inadequacies of the oral and written communication medium. A prototype, or mockup, system was proposed as an improvement to this medium and the manner of its use during the initial SDLC stages was discussed.

The desirability of modularity in the system program structure was discussed. This raised the issue of software toolkits which could adopt the role of particular system modules so that the developers could concentrate on the application rather than the supporting functions. The database was cited as an existing advanced software tool as are graphics libraries although these are much less advanced and operate at a lower level. Discussion centred around the user interface which consumed considerable development resources during the HEC LIS project. It was proposed, and supported by other authors, that a user interface toolkit is an urgent requirement for MIS and DSS development.

The CODASYL network and Relational DBMS were briefly compared with the intention of highlighting some common misconceptions regarding the advantages of the Relational model, in particular with regard to its application to LIS. Two extended formalism models, Extended Network and Post Relational, were considered and it was concluded that future LIS development, at least at the research or prototype level, should concentrate on implementation using either Post Relational or Extended Network models. The Post Relational model was suggested to be preferable because of its greater degree of data independence.

Chapter 5. CONCLUSION

5.1 Summary

The evolution of systems development methodology has been studied along with the literature which discusses the deficiencies of the traditional approach and proposals to improve it. The relationship between LIS development and MIS / DSS development has also been observed and clearly the example of the latter has influenced LIS.

The HEC LIS project has provided the opportunity for the formulation of another implementation strategy. This was carefully researched using the known deficiencies of previous methods and incorporates new techniques proposed by various systems development "experts".

Two methods emerged as being of particular benefit, namely:

- (i) Familiarisation Stage,
- (ii) Iterative cum prototype approach to program development.

5.1.1 Familiarisation

A new stage in the SDLC, Familiarisation, proved a very valuable exercise. The achievements of this stage were to quickly educate the consultant in the functions, objectives and methods of the organisation, prior to entering the formal stage of Requirements Definition. The most significant benefit, however, was establishing an excellent rapport between the consultant and users. This opened a most effective communication channel for the contribution of ideas and information.

5.1.2 Iterative Approach

The iterative approach allowed a departure from the usual situation, where users are not involved during program development [Cronan 1984]. This provided three important benefits:

- (i) User Commitment.

During this period users developed enthusiasm and loyalty toward

the system. They were able to mould system development through evaluation and requests for modification of each program version. The commitment grew stronger as users observed their own ideas embedded in the emerging system.

(ii) Fulfilment of requirements.

Continued system testing and modification during program development improved the assurance that it would fulfil user requirements.

(iii) Aid to training.

Users required very little additional training when the system became operational as a result of their earlier participation throughout program development.

5.1.3 Improvements to the Systems Development Life Cycle

The evaluation in chapter 3 identified the deficiencies of the methodology and chapter 4 proposed an improved approach in response to these problems. The resulting modifications introduced a prototype, or 'mockup', system during Requirements Analysis and Formulation to assist users. This provides many benefits of which the two most significant are:

- (i) improves the communication medium between developer and user,
- (ii) extends the users resources in their attempt to formulate future system and information requirements.

Modifications to the development methodology also include the use of a prototype approach during program development. This, however, is conditional on the use of a suitable 4GL, or similar development environment. Since such a 4GL is not currently available, it was therefore suggested that use of the iterative approach, adopted for the HEC project, be continued. Although the period between program versions is longer than when using a 4GL, this can be reduced by modularising the program and using toolkits for the user interface and graphics system development.

The HEC project has provided an actual environment within which to implement and evaluate the methodology. It has been possible to observe the performance of the various techniques and encounter, first hand, the typical problems that arise during design and development of a complex project. It has

also provided a context within which to analyse and evaluate other LIS research and literature.

5.2 Concluding Issues

From this experience the following concluding observations are formed.

"People Problems"

From the very early days of LIS development, indeed development of any information system, developers and researchers alike have sought a reason for system failures. It was reported earlier [Yourdon 1979] that programming of system specifications were first thought to be deficient, then system design, later user requirements were the culprit and most recently the communication between analyst and users is upheld as the cause of failure [Cronan 1984].

The HEC project not only confirms this conjecture, but contributes further insight into the true problem. The outcome of systems implementation is above any other factor, at the mercy of the strongest actors involved. This may be a single person, a cooperative, section, branch or an entire organisation. However, rather than a single dominant entity, there usually exists a number of domineering entities vying for control. A typical scenario is, that the losers of the struggle become non co-operative and obstructionist [Shrivastava 1984] and this is an underlying cause of the communication problem between system developers and the other actors.

Regardless of the scale of the project, LIS developments especially, will bring together people who have apparently conflicting views and feel opposed to one another. This may be caused by a number of factors, for example, a desire to protect personal or corporate interests, intrusion of job space, threat of discovering blunders or procedural inefficiencies [Graham 1981, Zwart 1981], but the inevitable result is a guarded level of co-operation, communication and contribution. However, a LIS implementation cannot succeed without a high level of each of the former ingredients from every actor involved [McLaughlin 1982b].

Therefore the task of the LIS developer is to bring together all actors and create an environment such that the suggested co-operative spirit prevails. To do this he requires highly developed political skills of negotiation, arbitration and conciliation in addition to a background in both

systems development and land information management. In other words, he must not only bring together the actors but also be in a position to assess their respective arguments and requirements from both technical and political perspectives while concurrently coercing them toward the ultimate common goal of sharing land information.

This thesis and associated case study has provided the opportunity for the author to observe and undertake a leading role in an actual LIS implementation. It appears that the system is successful, although from the point of view of this conclusion the assertion is irrelevant. The fact of importance is that an empirical study has shown that the influence and interplay of the various actors in LIS implementation and operation are the most significant influences in shaping the system and determining the outcome of its implementation. Allied with this discovery is the realisation that it is not technical project management but human management that is of greatest consequence.

REFERENCES

- Barney, C. (1983), "Software Houses Open Windows Wide for Microcomputers", Australian Electronics Engineering, Nov., p17.
- Bass, L. (1985), "A Generalised User Interface for Applications Programs", Communications of the ACM, Vol.28, No.6, pp617-627.
- Brady, L.I. (1985), "Entities, Relationships and the Universal Relation", Proceedings of 1st Pan Pacific Computer Conference, Melbourne, pp190-202.
- Breheny, M.J. (1984), "Policy, Implementation and Information: A Review of Changing Roles and Requirements", Proceedings of URPIS 12, Woolongong.
- Chiang, T.C. and Rose, G.R. (1983), "Design and Implementation of an E-R Database Management System", Entity Relationship Approach to Information Modelling and Analysis, P. Chen (editor), North Holland.
- Chilson, D.W. and Kudlac, M.E. (1983), "Database Design: A Survey of Logical and Physical Design Techniques", Database, Fall, 1983, pp11-15.
- Coleman, S. (1985), "Micro 4GLs Fall Short of the Mark", Computerworld, Vol.7, No.50, June 15th, p21.
- Condon, R.J. (1978), Data Processing Systems Analysis and Design, 2nd edition, Reston Publishing Company.
- Courtaz, J. (1985), "A Layout Abstraction For User-System Interface", Special Interest Group on Computer and Human Interaction, Vol.16, No.3, pp18-24.
- Cronan, T.P. and Means, T.L. (1984), "System Development: An Empirical Study of User Communication", Database, Spring 1984.
- Cronan, T.P. (1984), "Application Systems Development: A Communication Model for Business Users and DP Personnel", DataBase, Vol. 16, No.1, Fall 1984.
- Cullum, R.L. (1985), "Iterative Development", Datamation, OEM edition, Feb.
- Curley, K.F. (1983), "The Role of the Champion in DSS Implementation", Information and Management, Vol.6, No.4, pp203-209.
- Curtis, R.M. and Jones Jr, P.E. (1984), "Database: The Bedrock of Business", Datamation, June, pp163-166.
- Date, C.J. (1984), "How Relational Systems Perform", Computerworld, Vol.6, No.40, pp23-26.

- Date, C.J. (1982), An Introduction to Database Systems, 3rd edition, Addison-Wesely Publ.
- Date, C.J. (1983), notes taken during presentation of "Relational Database: An Overview", 10th Australian Computer Conference, Melbourne.
- Dee, D. (1985), "Developing PC Applications", Datamation, OEM edition, Feb.
- De Marco, T. (1978), Structured Analysis and System Specification, Yourdon, Inc., New York.
- De Marco, T. (1979), "Structured Analysis and System Specification", Classics in Software Engineering, Yourdon, Inc., New York, pp411-424.
- Dewhirst, M. (1984), "Putting Database Management Systems Into Perspective", Computerworld, Vol 6, No.30, Jan. 27th, pp17-19.
- Dickson, G.W. and Janson, M.A. (1984), "The Failure of a DSS for Energy Conservation: A Technical Perspective", Systems Objectives Solutions, Vol.4, No.2.
- Ditlea, S. (1984), "Befriending the Befuddled", Datamation, June.
- Divard, S. and Huff, S. (1985), "Users as Application Developers", Datamation, Vol.8, No.2.
- Ein-Dor et al, (1984), "Perceived Importance, Investment and Success of MIS, or the MIS Zoo", Systems Objectives Solutions, Vol.4, No2.
- Garnett, L. (1985), "Demands of 4GL Overtax System", Computerworld, Vol.7, No.51, June 21st.
- Ginzburg, M.J. (1981), "DSS Success: Measurement and Facilitation", Centre for Research on Information Systems Working Paper, No.20, NYU.
- Graham, R.J. (1981), "Understanding the Benefits of Poor Communications", Interfaces, Vol.11, No.3, June, pp80-82.
- Hinds, M. (1985), "Dispelling Some Blatant 4GL Myths", Computerworld, Vol.7, No.35, March 1st.
- Humphries, B. (1984), "Crisis in Land Management", Proceedings of Austra Carto One, Perth, August, pp5-17.
- James, B. (1985), "Your Choice Is Either Network or Relational", Computerworld, Vol.7, No.34, p24.
- Johnson, T.R. (1981), "Evaluation and Improvement of the Geographic Information System Design Model", Proceedings of Specification and Design of Geographic Information Systems, Kensington, NSW, Sept., pp199-240.
- Keider, S.P. (1981), "Why Projects Fail", Proceedings of Specification and Design of Geographic Information Systems, Kensington, NSW, Sept.
- Keyes, (1984), "Relational Database Still Lack Ability to Automate Efficiency", Computerworld, Vol.7, No.50, June 15th, pp22-23.

- King, D. (1984), Current Practices in Software Development, Yourdon Inc., New York.
- Land Information Systems Advisory Committee (LISAC), (1982), Utility Information Systems, Technical Report.
- Landry, M, Pascot, D., Briolat, D. (1985), "Can DSS Evolve Without Changing Our View of the Concept of 'Problem'? ", Decision Support Systems, Vol.1, No.1, pp25-36.
- Lemmons, P. (1983), "Microsoft Windows", Byte, Vol.8, No.12, Dec., pp48-54.
- Linstone, H.A. (1984), Multiple Perspectives for Decision Making, Elsevier Science Publ., New York.
- Love, W.R. (1982), A Developmental Land Information System, School of Surveying, University of Tasmania.
- Love, W.R. (1983), Some Aspects of Schema Design and Implementation for LIS, School of Surveying, University of Tasmania.
- Love, W.R. and Zwart, P.R. (1983), "Some Aspects of Database Schema For land Information Systems", Proceedings of URPIS 11, Brisbane, Nov., pp3A-13-22.
- Love, W.R. (1985), "A Microcomputer Based Systems for Managing Land Information", The Australian Surveyor, Vol.32, No.7, pp523-535.
- Lucas, H.C. (1978), "Empirical Evidence for a Descriptive Model of Implementation", MIS Quarterly, Vol.2, pp27-41.
- Martin, J. (1981), Application Development Without Programmers, Savant.
- McCosh, A.M. (1984), "Factors Common to the Successful Implementation of Twelve DSS and How They Differ from Three Failures", Systems Objectives Solutions, Vol.4, No.1
- McEnaney, M (1985), "4GLs No Substitute for Cobol, Says IDC", Computerworld, Vol.7, No.33, Feb. 15th., p8
- McLaughlin, J. (1982a), Implementating Land Information Systems Seminar, School of Surveying, University of Tasmania, Nov, 1982.
- McLaughlin, J. (1982b), "Land Information Management: A Canadian Perspective", Proceeding of URPIS 10 Conference, Sydney, pp212-229.
- McLaughlin, J. (1982c), Notes taken during Land Information Systems for State and Local Government Seminar, School of Surveying, University of New South Wales.
- Mumford, E. and Henshall, D. (1979), A Participative Approach to Computer Systems Design, Associated Business Press, London.
- Naisbitt, J. (1984), Megatrends, reviewed in Computerworld, Vol.6, No.43, pp15-19.

- Nijssen, G.M. (1983), "Intelligent Information Systems and the Community",
Proceedings of URPIS 11, Brisbane, pp2,1-7.
- O'Connor, R.J. (1983), "Interfacelift", Australian Microcomputer, May/June,
pp14-15.
- Oppermann, R. (1984), "User Participation: Some Experiences and
Recommendations", Systems Objectives Solutions, Vol.4, No.3,
August, pp157-168.
- Patkin, M. (1984), "Software Ergonomics: The Quest to Define Values of
User-Friendliness", Computerworld, Vol.6, No.43, April 27th, Patkin,
pp30-31.
- Ramnath, R. (1983), "Application Transfer Teams - Key to Implementing Land and
Facilities Information Systems", Proceedings of URPIS 11, Brisbane,
pp3B20-26.
- Richardson, G.L., Butler, C.W., And Tomlinson, J.D. (1980), A Primer on
Structured Program Design, Petrocelli Book, Inc.
- Ross, D.T. and Schoman Jr, K.E. (1979), "Structured Analysis for Requirements
Definition", Classics in Software Engineering, Yourdon Inc., New York,
pp365-386.
- Rund, D.L.S. (1982), Contributing author to Teorey, T.J. and Fry, J.P. (1982),
Design of Database Structures, Prentice-Hall, New Jersey, pp97-130.
- Saja, A.D. (1985), "The Cognitive Model: An Approach to Designing the
Human - Computer Interface", SIGCHI, Vol16. No.3, pp36-39.
- Sanders, G.L. (1984), "MIS/DSS Success Measure", Systems Objectives Solution,
Vol.4, No.1
- Scharer, L. (1981), "Pinpointing Requirements", Datamation, pp139-151.
- Schewe, C.D. (1976), "The Management Information System User: An Exploratory
Behavioural Analysis", Acadamey of Management Journal, Vol.19, No.4,
pp557-590.
- Scott, P. (1985), "4GL Buyer Beware", Computerworld, Vol.7, No.38, March 22nd.
- Segall, M.J. (1984), "The Use of Prototyping to Aid Implementation of an
On-Line System", Systems Objectives Solution, Vol.4, No.3, pp141-156.
- Shrivastava, P. and Etzar, M. (1984), "A Decision Support System for Strategic
Marketing Systems", Systems Objectives Solutions, Vol.4, No.3
- Simpson, G. (1983), "Fourth Generation Languages or a Computer Revolution?",
Proceedings of 10th Australian Computer Conference, Melbourne,
pp728-734.
- Spock, R.E. (1985), "Break With Tradition", Datamation, May.
- Stamen, J. and Costello, W. (1985), "Decade-old Debate Still Raging Over
Practicality", Computerworld, Vol.7, No.34, pp24-25.

- Steele, P.M. (1983), "What is needed? Data Dictionaries/ Data Directories, 4G1 or Conceptual Schema?", Proceedings of 10th Australian Computer Conference, Melbourne, pp735-746.
- Stephens, W. (1984), "Application Transfer Teams (ATT): Key to Implementing Land and Facilities Information Systems", Proceedings of 12th International Cartographic Association Conference, Perth, August, pp715-730.
- Strehlo, K. (1984), "Environment Software: Opening New Windows on Your Work", Personal Computing, Vol.8, No.2, Feb, pp107-113.
- Sullivan, K. (1985), "Thumbs Down on Solution", Computerworld, Vol.17, No.51, June 21st, p23.
- Sweet, F. (1985a), "Painting a New Picture", Datamation, May.
- Sweet, F. (1985), "Managing Data-Driven Development", Datamation, June.
- Tay, B.W. and Kinoshita, E. (1983), "The Impact of Fourth Generation Languages on the Data Processing Environment", Proceedings of 10th Australian Computer Conference, Melbourne, Sept., pp747-795.
- Teorey, T.J. and Fry, J.P. (1982), Design of Database Structures, Prentice-Hall, New Jersey.
- Tesler, L. (1985), "Legacy of the Lisa", MACWORLD, Vol2. No.9, Sept., pp17-22.
- Wenig, R.P. (1982), "Systems Analysis Checklist", Systems Development Management, Auerbach Publ. Inc., New York, pp59-66.
- Wetherbe, V. (1984), "MIS Research : A Profile of Leading Journals and Universities", Database, Vol.16, No.1.
- Yestingsmeir, J. (1984), "Human Factors and Considerations in Development of Interactive Systems", SIGCHI, Vol16, No.1, pp24-27.
- Young, T.R. (1984), "Superior Prototypes", Datamation, May 15th, pp153-158.
- Yourdon, E. (1979), Classics in Software Engineering, Yourdon, Inc., New York.
- Zwart, P.R. (1981), "Administrative Reform for Land Information Systems", Proceedings of URPIS 9, Geelong, 1981.
- Zwart, P.R. (1984), "The Rise and Decline of Land Information Systems", Proceedings of 12th International Cartographic Association Conference, Perth, pp123-133.

APPENDIX A

Familiarisation Exercise

A familiarisation exercise was conducted to inform the consultant of the operation and functions of the Survey section. This was achieved during a four week period involving extensive interviews, discussions and technical tours.

The following documents the information gleaned from the various sessions.

2. Land Administration Section

The Land Administration Section (LAS) is the section within the Commission which deals with the acquisition, management and disposal of Commission land. The LAS may not necessarily initiate a land dealing, it is however immediately notified of an intention to acquire or to dispose of land and will proceed to execute the necessary negotiations and transactions.

2.1 Filing Procedures Involved in Purchasing Land.

The Commission can secure ownership of land under any of three processes, namely:

- (a) normal vendor purchaser negotiations involving persons representing the Commission and the property owner.
- (b) compulsory acquisition of land under the Public Authorities Land Acquisition Act 1949.

(c) lands vested in the Commission under section 35 of the Hydro Electric Commission Act 1944 which (i) sets aside land for the use of the Commission (ii) may grant that land to the Commission if required.

Once a dealing has been initiated a LAS action file is opened. This file contains all outward correspondence related to the land parcel. Originals of all outward and inward correspondence are forwarded to the Records Section who maintain a complete, permanent file. This file has a unique ADEPT reference number. The ADEPT file references concerning the purchase and disposal of land are:

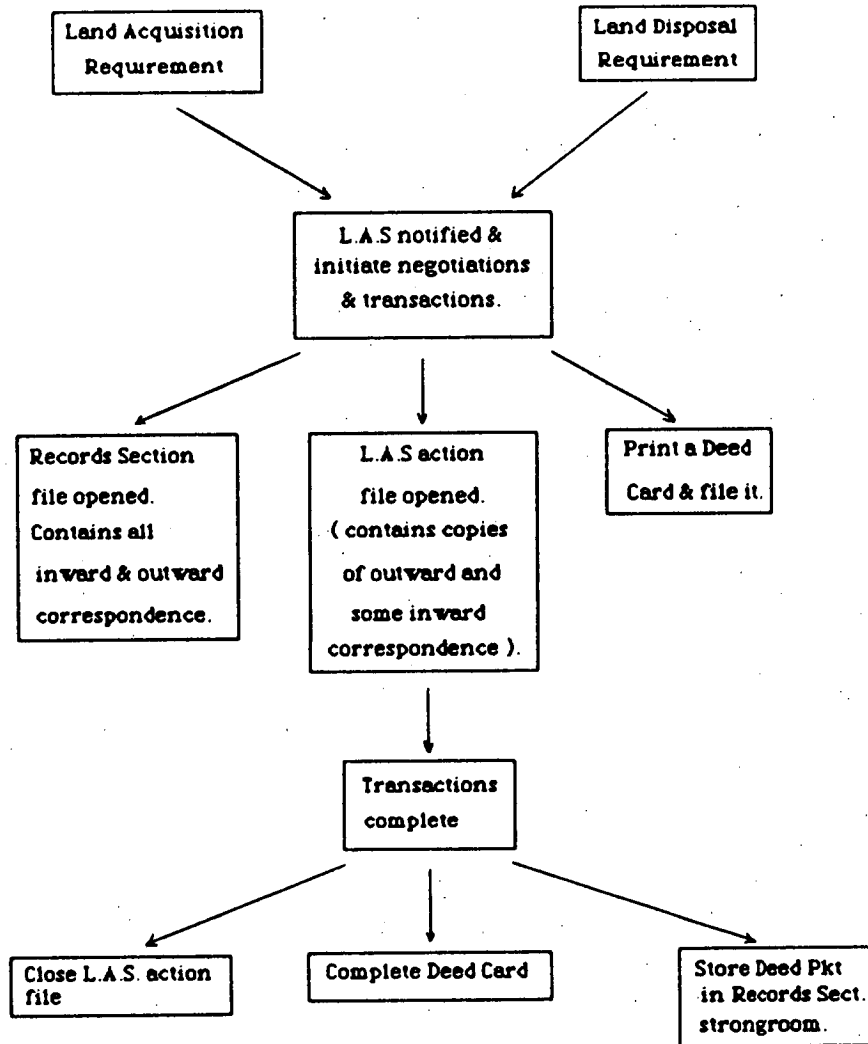
- AD - Administration, Disposal sale of property,
- DR - Development, purchase of Real Estate.

These classifications are both indexed alphabetically according to the name of the property owner.

After the action file has been opened a "deed card" is printed for the land parcel. This holds all information currently available concerning the parcel, such as, owner, locality, area. Later, when all transactions are complete, the remaining details, such as purchase price, C/T number, etc are filled in. The card is filed in the Deed

Card Box according to the municipality of the land parcel.

The deed cards are strictly an LAS index into the Commission filing system. They do not contain a description of the current legal or physical status of the land or its history. This information and details concerning the final settlement can be gleaned from the deed packets or Records Section file.



2.2 Land purchased for Distribution Branch Substations.

There are three classifications, namely:

- (a) Substation installed explicitly for the consumer.

The consumer buys the plant and makes a suitable piece of land available (without charge).

In this situation the land is rarely transferred to the Commission. The agreement is negotiated between the Commission and the consumer, however, it is often not formalised.

(b) Substation installed primarily for the consumer.

The Commission supplies the plant and the consumer makes land available for lease (although the Commission prefers to purchase). Often the Commission will install a transformer for the consumer and another for its own use, i.e. to supply the neighbouring region).

(c) Substation for Commission use.

The Commission either purchases or leases (prefers to purchase) land for substation site. This usually always involves a formal process whereby the dealing is registered at the LTO and a title is issued.

There are many cases where agreements have been negotiated with original owners but not formalised. From time to time these cause problems when the land is subdivided or sold and the new owner challenges the Commission's presence on the land. This usually results in the Commission negotiating a lease or purchase of the site, e.g. Eski Ice Co. There are probably about one hundred sites in the state where the interest has not been transferred to the Commission although approval was given by the original consumer.

2.3 Wayleave Agreements.

2.3.1 Description.

A wayleave, or wayleave easement, is an easement secured between the Hydro and a property owner giving the Hydro an interest in his land for the purpose of power transmission. The means of transmission may be overhead lines or underground cables. The easement is also restrictive in that it prohibits certain activities within its dimensions. Each agreement is accompanied by a plan of the property and the transmission line.

2.3.2 Wayleave Enquiries.

Wayleave enquiries generally arise from Solicitors enquiring on

behalf of a client with regard to the existence of Commission encumbrances or intention to acquire an interest in the clients land. Such enquiries are presently received at the rate of about 30 per day.

These enquiries are often answered with a standard reply.

To determine whether or not a property is affected it must be geographically located on a map which shows all Hydro transmission lines. If, when located, the parcel is found to be in close vicinity of a transmission line then it is most likely affected by a wayleave agreement and will be recorded in the Wayleave search index. This is a two stage index which firstly lists all transmission lines in chronological order of their establishment, secondly it lists, in alphabetical order, all property owners on that transmission line with the wayleave agreement ADEPT file reference and file number.

The ADEPT system for transmission lines is as follows:

DT 85-103-18

"Development" file

Subject: Transmission lines

Line name,

property owner's name.

There are several difficulties in using this system. The foremost of these is spatially locating the parcel. Usually this is done from a title plan supplied by the enquirer. Title plans are notoriously abismal for the lack of detail shown so the task of locating the title plan on a map can be very difficult.

Often the current title can be identified as part of an original property affected by a wayleave. The problem then becomes one of solving whether or not the current title is affected by the same wayleave.

At present the LAS uses any map or chart that will assist them in identifying the parcel. In general this includes, 1:25, 000, 1:100,000, county charts, Master Planning Authority charts, Hobart and Launceston street atlases. The county charts are particularly important since the LTO charts of original grants is based on this series. Master Planning Authority charts are particularly useful in urban areas since these charts identify street numbers.

Other aids used in identifying properties are aerial photographs and Transmission line towers numbers. Transmission line towers are uniquely numbered and generally recorded on the wayleave agreement. This method can be misleading if the original line has been replaced since the new towers, although in a similar location, will have a different number.

Survey plan and lot number are often quoted on the supplied title plan. This would be a useful property identifier to assist in the search.

Some data on the Wayleave Register will become out of date, such as the title reference and owner's name, however this original information remains important.

2.3.3 Formalisation of Wayleave Easements.

Under the terms of its act (Hydro-Electric Commission Act 1944) the Commission is compelled to maintain a register of wayleave easements. This is essential since it is not mandatory for this interest to be recorded on land titles.

The Commission is presently in the process of formalising the wayleave agreement. These agreements have evolved from a particularly informal agreement originating, it is suggested, from a period when men were proud to have an H.E.C. installation on or across their property. This was seen as participating in the development of the State. In the early years (1914, the Hydro-Electric and Metalurgic Co., 1915 - 1929 the Hydro-Electric Department, 1929.. Hydro-Electric Commission) until about 1940, wayleave agreements recorded few of the physical details of the easement. The information generally recorded was :

- (i) consideration
- (ii) property owner's name
- (iii) parish/municipality and the date of the agreement.

Important information omitted was:

- (i) C/T , deed or conveyance reference
- (ii) width of the easement and
- (iii) details of the restrictive nature of the easement.

A sketch of the easement was usually included however this depicted very little detail and was never shown to scale. One useful detail shown, however, was the transmission line tower number.

About 1940 provision was made in the wayleave agreement form for the C/T, deed or conveyance reference. Also the standard of sketch seemed to improve although it was still not a scale plan. About 1957 provision was made on the form for the inclusion of a clause covering the restrictions imposed by the easement.

During the early 1960's the wayleave agreement form was reprinted (changed colour from pink to white) and the restrictions clause was enhanced to include the width of the easement from the transmission centre line. By the mid 1970's a survey plan was attached to the agreement replacing the sketch diagram.

Since 1980 the process for securing an easement has substantially altered. It is now secured under the Public Authorities Land Acquisition Act 1949. The purpose of this is to ensure:

- (a) Each property owner is treated on an equal basis and on the same time scale. In practice this means that the notice to treat is served on all affected owners at the same time and interest on consideration is payable from the same date.
- (b) A proper survey of the easement is conducted and recorded on the title.
- (c) Registration of the easement on the owner's title and that of any other person with a legal interest in the land.
- (d) Issue of title for the easement to the Commission.

NOTE: The new Sheffield-Pieman transmission line is caught between the two systems. It employs an enhanced version of the wayleave agreement under section 48 of the Hydro-Electric Commission Act 1944 which gives the Commission the right to take the land for its own use and to take an easement if necessary. That is, the owners have agreed to grant a right for the easement to be negotiated by way of transfer sometime in the future.

2.3.4 Salient details on the Wayleave Agreement

- (a) consideration,
- (b) width of easement,
- (c) parish/property name/municipality,

- (d) C/T, deed or conveyance reference,
- (e) date of the agreement,
- (f) owner's name,
- (g) transmission tower numbers (from the diagram).

In the new instances where title issued for the wayleave the salient details are:

- (i) owner's name,
- (ii) date of registration of the easement,
- (iii) parish/property name/municipality,
- (iv) C/T reference,
- (v) width of the easement may be interpreted from the attached survey diagram,
- (vi) plan reference.

2.3.5 New system for wayleave identification.

This is a system for updating the original properties affected by wayleave to the current title reference and owner.

Generally the R.P. or G.L. reference has been extracted from the original wayleave agreement and the current title searched by the LTO. From the title plan parcels are plotted on the 1:25 000 map series and also on the aerial photography covering the line. The photograph coverage of each photo is also plotted on the map.

The titles are filed in their order of occurrence on the transmission line.

In regions where the Lands Department have supplied PID numbers for the plotted properties a recent title reference can be immediately identified from the Lands Department printouts. This makes the task relatively simple. The only difficulties remaining in this case are to identify and locate:

- (1) properties subdivided since map publication and
- (2) properties with erroneous title references for PIDs.

Wayleave agreements formed prior to the early 1940's typically omitted the C/T, deed or conveyance reference. In this case the current titles are found by plotting the transmission line on a county chart and tracing the title from original grants intersected. This method is also used where:

- (1) parcel boundaries are not plotted because of scale and,
- (2) where a 1:25 000 sheet is not available.

Reliability of the Lands PID to C/T reference is estimated at about 85%.

The later transmission lines (at least since mid 1970's) have reasonably accurate plans detailing the parcel, transmission line, easement, tower locations and numbers. It is thought that tower co-ordinates are maintained somewhere in the Commission. These could be used to bring wayleave plans onto the AMG. Those parcels traversed by the transmission line but not having towers can be brought in using lease squares transformations.

The Distribution Branch also secure wayleave easements for the lower voltage distribution power lines. The LAS is not particularly concerned about these although the drafting section (Craig) is concerned. Prior to 1980 the agreement for distribution wayleaves have been particularly rough, frequently not even recording the C/T reference. Since this time the LAS have moved to the Survey Section whereupon a detailed survey plan of the easement is now included in the agreement. The C/T and other important information is also recorded.

3. Aerial Photography Files.

3.1 Aerial Photography Flight Log.

This is a record of the flight number (unique identifier) with various other technical information regarding the photography, including details of the flight path (source location to destination).

3.2 1:100,000 Map Reference Flight Plan Folder

Using the 1:100,000 map series the aircraft flight path is plotted showing the photography coverage. This has been accomplished for films beyond film No. 162, dated 30/3/1976.

3.3 There is a file of all photography undertaken by the Commission from year dot. This file is indexed by subject (geographical).

3.4 Infra Red Photography

Infra red film is numbered and stored in a folder. A 1:100,000

Flight Plan folder is also maintained.

3.5 Lands Department and HEC Prints

Involves a variety of aerial photography from both Lands and the HEC. It is not indexed (or filed).

5.0 Survey Drafting Section

The function of the Survey Drafting Section with respect to land administration is to provide the relevant survey plans and field notes for the Survey and Land Administration sections. This is generally to support a land transaction or enquiry.

The appropriate plans are not necessarily held within the Commission. It is estimated that 80% of information supplied to support the above functions is obtained from outside the Commission. The principal source is the Registrar Generals Department and Lands Department. Others include Mines Department, Forestry Commission and virtually all other Government, semi Government and local Government agencies.

The Drafting section files Survey information in the Survey file. This system is indexed by a number generated from a field book and page number relating to a particular job. Access is gained to the system by locating the relevant land parcel on the 1:25,000 map series. The survey file index number is marked along side the parcel.

The file contains:

- (1) the field book and page number of the job relating to a particular parcel,
- (2) Survey request form,
- (3) Copy of the final (registered plan) and the field notes,
- (4) All reference material used to compile the plan.

It is suggested that the computer LIS contain the Survey File data. Reference material references could be held either on the computer Survey file, the field book or the final plan. In any event the recording of these references would allow the raw material to be destroyed thus preventing further physical storage problems of these plans.

The computer Survey file would contain the following:

- (1) Field book number and page number (file index)
- (2) Registered plan number and title number,
- (3) Date of the survey,
- (4) 1:25,000 map series sheet name and reference number,

Other data required (already included for LAS) is:

- (5) Deed packet number,

Survey file index numbers are of the format dddd-dd.

Reference material numbers are of a varied format expressed by aaaaaaaa. It is estimated that the a parcel could have, on average, 10 plan references. This amounts to some 80 bytes per parcel.

APPENDIX B

Resource Inventory

CONTENTS

Section	Page
1. SOFTWARE RESOURCES	
1. MS-DOS Operating System	1
2. MS-DOS Toolkits	1
3. Language	1
4. Database	2
5. Prototype Software	2
6. Software Summary	3
7. Budget for Recommended Software	4
2. HARDWARE RESOURCES	
1 Sirius 1 Microcomputer	1
1.1 System Specification	1
1.2 Sirius Workload	1
2 Epson MX-100 III Printer	2
3 Conclusions and Recommendations	2
3.1 Purchase of Additional Microcomputer	2
3.2 Enhancement of Processing Power	3
3.3 Memory Expansion	3
3.4 Physical Storage	3
4 Budget for LIS Development Recommendations	4
5 Summary of Hardware Resources and Workload	5

SOFTWARE RESOURCES

1. MS DOS Operating System

MS DOS is the widely distributed Microsoft (TM) operating system for the 16 bit micro computer range. It is the principal operating system of at least 30 micro computer systems. The current version 2.1 is not yet available for the Sirius although the beta release (2.08) has been acquired. MS DOS was purchased in August 1983 with the Sirius computer.

2. MS DOS Toolkits

Three toolkits are available for MS DOS. Essentially these comprise program development support utilities. Unfortunately toolkits for MS DOS version 2.1 are not yet available.

3. Language

There are two languages, Pascal and C, suitable for LIS program development. The Job Inventory System program utilises Microsoft (TM) Pascal, initially version 3.04 owned by the University of Tasmania. This version, however, has some unresolved "bugs" as well as severe technical limitations which restricted program size. Because of the severity of these deficiencies it was recommended that the Hydro purchase a copy of Pascal direct from Microsoft thus acquiring the new version, 3.13, in order to successfully complete the Job Inventory System. The HEC is now a registered owner of Microsoft Pascal and is entitled to compiler upgrades according to the usual upgrade arrangement.

The limitations of the Pascal compiler were known from the outset hence it was resolved, prior to the commencement of the JIS, that the LIS be developed using the C language. The C compiler was ordered (by the University of Tasmania) and received. The C interface for the database was also ordered. Documentation subsequently received with Pascal 3.13 identified and resolved the "bugs" in the earlier version and relieved the program size limitations. Follow up enquiries by telephone to Microsoft Australia (Sydney) revealed that a further release, due at the close of April 1984 would lift all previous program size restrictions. On the strength of the version 3.13 documentation and announced further enhancements it is recommended that Pascal be adopted as the LIS development language. The C interface order has subsequently been cancelled.

4. Database

The LIS utilises MDBS III database software. This is a CODASYL extended network database and represents the most sophisticated database software available for a micro computer.

Important features of MDBS III are:

- (i) Data types supported: Integer, Unsigned, Real, Idec, Char, String, Date and Time.
- (ii) MDBS supports all set relationships, that is, 1:1, 1:N, M:1, N:M and recursive sets.
- (iii) Record indexing available is, FIFO, LIFO, sorted AZ or ZA on one or many data items with user definable index key width, and hashing.
- (iv) Data dictionary and data definition language.
- (v) Programable through a host language.
- (vi) Read/write security at the record level.
- (vii) Record clustering and physical database partitioning support.
- (viii) Optional software packages including RTL (recovery transaction logging), QRS (query language) which are available if required.
- (ix) No impractical physical restrictions.

5. Prototype Software

It is intended that a relational database management system (RDBMS) marketed by Micro Database Systems as "KNOWLEDGEMAN" be purchased. This system will be used to develop a prototype system to computerise the Land Administration and associated filing systems. The prototype system will be operational during the LIS development period providing personnel with introductory computer experience.

This package also incorporates a powerful spreadsheet capability which can provide a simple accounting system for the Survey Section.

KNOWLEDGEMAN will be purchased by Applied Information System Pty. Ltd. and licensed to the HEC.

5. Software Summary as at 27th April 1984

<u>Name</u>	<u>Description</u>	<u>Cost</u>
Microsoft (TM) DOS	Operating system.	Version 1.25 H
MS DOS	Current version 2.08 (beta) Owned by HEC and entitled to update version 2.1 and subsequent versions for update fee.	(supplied with Sirius) Version 2.08b courtesy of Management Technology
MS DOS Hard Disk Toolkit	Support utilities for MS DOS 2.08H Owned by HEC and entitled to updated version for MS DOS 2.1 and subsequent versions for update fee.	\$325.00
MS DOS Programmer's Toolkit	Programmer's utilities, e.g. PMATE, MODCON. Owned by HEC and entitled to update version for MS DOS 2.1 and subsequent versions for update fee.	\$325.00
MS DOS Graphics Toolkit	Graphics support, character sets, Character font generator. Owned by HEC and entitled to update version for MS DOS 2.1 and subsequent versions for update fee.	\$325.00
Microsoft (TM) Pascal	LIS application development language. Current version 3.13 (large memory module). Owned by HEC and entitled to subsequent versions for update fee.	\$750.00
		<hr/> \$ 1,825.00 <hr/>

6. Budget for Recommended Software

<u>Name</u>	<u>Description</u>	<u>Cost</u>
License fees:		
MDBS III 3.06a	Extended CODASYL network database. Owned by University of Tasmania and Licensed to HEC.	\$2,500.00
KNOWLEDGEMAN	Relational Database Management System, Spreadsheet and Work Processor. Owned by Applied Information Systems Pty. Ltd. and licensed to HEC.	
Software upgrades:		
MS DOS 2.1	Upgrade from 2.0b to 2.1	100.00
Hard Disk Toolkit	for MS DOS 2.1	100.00
Graphics Toolkit	for MS DOS 2.1	100.00
Programmer's Toolkit	for MS DOS 2.1	100.00

HARDWARE RESOURCES

1. Sirius 1 Micro Computer

1.1 System Specification

The Sirius 1 is one of the many 16 bit MS-DOS based micro computers. The H.E.C. computer has a 10 Mb in built hard disk and a single 600 Kb floppy disk drive. This offers a significant performance increase over the dual floppy disk based version. The central processor unit (cpu) is the Intel 8088 micro processor chip operating at 5 MHz. Two serial and one parallel communications ports offer the ability for printer, modem and network support. Sirius has the highest screen resolution (800 x 400 pixels) of all MS-DOS machines. This combined with the long persistence green phosphor screen provide an excellent operator work environment, minimising the potential for operator stress and eye strain. It also offers an excellent opportunity for graphical display of land parcels.

Four expansion slots available which may be used, for example for additional memory, network interface or additional cpu.

The system is supplied with 256 Kb RAM and may be expanded to 896 Kb.

1.2 Sirius Workload

This Sirius was purchased in September 1983 to run the custom MOSS Formatting Package and act as a terminal to CEANET's Melbourne based PRIME 750. A Visual 550 dumb terminal has been purchased to take over the PRIME terminal work and only in the event of emergency jobs or the processing of a TC1 tape is the Sirius required to assist in this role. The frequency of this interruption cannot be estimated since the occurrence of "rush" jobs are sporadic and totally unpredictable. It is certain that a second terminal to the PRIME will be necessary from time to time.

A new task for the Sirius is to operate the recently installed Job Inventory System. This system, currently in the data acquisition stage, is in operation approximately 2 - 3 hours per day. It is anticipated that this will diminish to approximately 1 - 1½ hours per day for daily updating and enquiries when data entry is complete.

2. Epson MX-100 III Printer

The Epson MX-100 III is a dot matrix 132 column text/graphics printer. It has both serial and parallel ports with a 512 Kb print buffer built in to the serial port. It was purchased in January 1984 to provide an output mechanism for the Moss Formatting Program. It is presently used for MOSS file dumps and will soon be required for Job Inventory System enquiry printouts.

3. Conclusions and Recommendations

3.1 Purchase of additional micro computer

During the development of the Job Inventory System it became apparent that the Survey Section hardware resources were insufficient. Once the JIS became operational the single computer could not service the requirements of both development machine for the LIS and operational machine for the JIS. It was therefore proposed that a second Sirius be purchased. Its purpose is to operate the JIS and act as a second PRIME terminal when required. This proposal is now under consideration by the Commission.

3.2 Enhancement of Processing power

Experience in large application development using the Sirius micro computer shows that approximately 30 - 35% of time during programming is spent in program compilation. This amounts to 3 - 3½ hours per day or approximately 330 hours (about 34 working days) over a 4 month program development period. Whilst other activities can be performed during program compilation a low level of productivity is achieved since there is, on average, only 15 - 30 minutes available. It is desirable therefore to reduce compilation time.

The duration of compilation is primarily determined by the performance of:

- (1) the language compiler and (2) the cpu.

The language compiler is discussed in the Software Resources section.

The Sirius cpu performance can be increased by introducing a more powerful processor via the expansion slots. There is, at present, only one suitable processor commercially available for this task. It is the Intel 8086 micro processor chip. The product, called

"PC EXPRESS", is marketed with $\frac{1}{2}$ Mb (512 Kb) RAM and operating at 10 MHz. RAM disk software is supplied for no additional cost. The total cost (less 20% sales tax, less negotiated discount \$180) is \$2,000. The estimated performance gain is 200% - 300%. This gain would be reflected in both the LIS development time and operational performance.

It is recommended that this product be purchased for the LIS development computer.

3.2 Memory expansion

Micro computer applications development experience also indicates a requirement for a large amount of RAM, in particular for the development computer. A minimum of $\frac{1}{2}$ Mb is advisable. This requirement would be met by the proposed purchase of the PC EXPRESS expansion board.

A mandatory 384 Kb is required to operate the JIS. This represents a 256 Kb memory upgrade for the floppy disk based Sirius. Memory expansion is available in 128 Kb upgrades which may be purchased as single boards or "socketing" 128 Kb capacity rows of either a 384 Kb or $\frac{1}{2}$ Mb (512 Kb) expansion boards.

It is recommended that a 384 Kb capacity expansion board be purchased for the Sirius operating the JIS and 32 x 8 Kb RAM chips be purchased to fully expand it to 384 Kb providing this machine with a total $\frac{1}{2}$ Mb RAM.

The cost of the board with 128 Kb resident RAM is \$700. The further cost of 32 x 8 Kb RAM chips @ \$11.00 each is \$352. The total cost is \$1052.

3.3 Physical Storage

It is anticipated that the total storage requirements for the LIS will not exceed the 10 Mb capacity of the hard disk system. However, it is possible that the specification of additional data required by Land Administration or other Sections may exceed 10 Mb. This should not cause any difficulty since third party manufacturers supply Sirius compatible hard disk systems which operate through the serial port. The Sirius mother company has announced a 40 Mb hard disk. Price is unknown.

The above recommendations are for system development only. Various LIS implementation strategies emerging from requirements formulation

and system design phases may require additional resources.

4. Budget for LIS development recommendations

Item Description	Quantity	Price/unit	Total
Sirius 1 micro computer 2 x 600 Kb floppy disk drive system, 128 Kb RAM	1	\$3,750	\$3,750
PC EXPRESS 8086 cpu expansion board, 10 MHz clock, 1/2 Mb RAM, RAM disk software	1	2,000	2,000
384 Kb memory, expansion board, 128 Kb RAM resident	1	700	700
8 Kb RAM chips	32	11	352
			<hr/> \$6,102 <hr/>

5. Summary of Hardware Resources and Workload

The hardware resources of the Survey Section as of 27th April, 1984 are as follows:

Description	Specifications	Value	Workload		
			Number of Tasks	Average time per task	Frequency of use
Sirius 1 Business micro computer	8088 16 bit CPU, 5 MHz clock 2 serial and 1 parallel comm port, 10 Mb hard disk, 600 Kb 5½" floppy disk drive, 800 x 400 screen resolution 256 Kb RAM	\$9,300	2	8-14 hrs/day 6 days/week	Mar-Dec 84
Epson MX-100 III Dot matrix Printer	132 column text/graphics printer 1 parallel, 1 serial port, 512 Kb print buffer	\$1,500	3	1-4 hrs/day 3-5 days/week	Mar-Dec 84

APPENDIX C

Implementation Report

CONTENTS

Organisation Of This Report	(i)
Introduction	(ii)
Chapter 1. Identification of Problems	1
Chapter 2.. Ideal System Characteristics	7
Chapter 3. System Applications	10
Chapter 4. Data Definition Table	13
Chapter 5. Acceptance Criteria	21
Chapter 6. Implementation Proposal	24

CONTENTS

Organisation Of This Report	(i)
Introduction	(ii)
Chapter 1. Identification of Problems	1
Chapter 2.. Ideal System Characteristics	7
Chapter 3. System Applications	10
Chapter 4. Data Definition Table	13
Chapter 5. Acceptance Criteria	21
Chapter 6. Implementation Proposal	24

INTRODUCTION

A rigid control plan and formal database design are an essential component for a successful Land Information Systems (LIS) implementation. The Hydro Electric Commission (HEC) LIS project proceeds along the guidelines of the strategic plan for design and implementation as set out in the Implementation Plan.

The strategic plan comprises nine fundamental stages, namely:

- (a) Investigation and understanding of the organisation's functions and the processes involved in performing these.
- (b) Analysis of the operational aspects of the organisation and identification of inefficiencies and deficiencies in the processes. Determination of the nature and cause of these problems and how they could be overcome.
- (c) Investigation and compilation of an inventory of resources available to contribute to the development of a system designed to overcome the problems identified.
- (d) Proposals for alternative system scenarios.
- (e) Selection of the best scenario or compromise.
- (f) & (g)
On the basis of the stated investigation, to design an information system (database and application programs) to satisfy the information requirements of the organisation.
- (h) Production and evaluation of a prototype system.
- (i) Refining the system.

The following report represents a summary of stages (a) through (c) and concludes with the presentation of stage (e), i.e. recommendations for the implementation of a LIS.

ORGANISATION OF THE REPORT

The report is organised into six chapters, the first five of which are concerned with summarising various issues raised during the requirements formulation stages. Chapter 1 summarises the problems presently encountered in performing Survey Section functions. Chapter 2 details the characteristics intending to overcome existing problems. Chapter 3 briefly describes the specialist application requirements for the system. That is, applications other than piecemeal retrieval of land parcel data. Chapter 4 is concerned with detailing the data to be included in the system and their characteristics. A description of success indicators and acceptance criteria is given in chapter 5.

The final chapter, chapter 6, forwards the recommendations for implementing the LIS. Two implementation scenarios are described and using the method of advocacy planning a final system is proposed.

Chapter 1. IDENTIFICATION OF PROBLEMS

The functions of the Survey Section and Land Administration Sections are numerous, varied, and usually involve a degree of urgency. Both sections are pioneers of Commission undertakings since few, if any, new developments can proceed before either an investigation survey has been conducted, or the necessary legal negotiations and transactions are completed. Requests for information or work to be undertaken evolve from within the Survey Section, other Commission branches and the Public. Typically all requests have a common characteristic of a high, or urgent priority.

This chapter summarises the major problems and deficiencies identified in the existing systems for processing work in the Survey and Land Administration sections. These problems were expressed mainly through the two questionnaires and interview sessions.

C 1.1 Problems Identified

C 1.1.1 Lack of Staff

The Land Administration Section (LAS) consists of the Chief Surveyor, Chief and Assistant Property Clerks and a Clerk. Three persons from the Survey and Drafting sections are capable of relief work. It is observed that efficient work performance in the LAS requires a large degree of "local" knowledge. For example, this involves a knowledge of current and past land dealings, the location of Commission land and installations and a thorough understanding of the ADEPT filing system. The estimated lead time to acquire this knowledge is one and a half to two years.

It is generally considered that the LAS, in particular, is considerably under staffed. It is thought that a key factor in seeking approval for increasing the staff level may be the qualification and quantification of job requests provided by an analytical / statistical component in the job inventory.

C 1.1.2 Job Inventory

Until recently, incoming job requests have not been effectively monitored or recorded, as a result some requests have been lost or forgotten. Most importantly, an ad-hoc system such as it was, does not provide a complete picture of workload or work progress thus impeding management's ability for effective administration.

C 1.1.3 Inability to Monitor Job Status

The state of completion of requests is a frequent enquiry from within and without the Survey Section. For example, "Is the job in progress?", "At what stage of completion is it at?". "Are the survey plans complete?". "Have the plans been lodged at the Titles Office?".

Many job requests are processed by different people at various stages thus making this information more difficult to obtain. Personal enquiries are the most effective means of determining job status, however, it is reported that this results in frequent, irritating and unnecessary interruptions to LAS staff.

C 1.1.4 Non-Assignment of Priority to Job Requests

Often work requests are submitted with no apparent preference for priority. An indication of this would assist those carrying out the request in order to plan their work more effectively.

C 1.1.5 Common Filing System

Presently a common filing system for recording land information, job requests, map, photography and file references does not exist. The typical media for communication of this information is verbal. This is prone to misinterpretation and failure.

C 1.1.6 Un-Registered Land Interests

There are situations where the Commission has installations on, or

traversing, private property for which the legal interest is either not registered or non-existent. The problem is magnified by the fact that the existence and whereabouts of such interests are generally unknown. In many cases the problem is identified when property ownership is transferred and the new owner forwards an enquiry regarding the installation.

C 1.1.7 Facility to Peruse Records

In many cases it is presently time consuming to locate and peruse a single file record. It is necessary to visit the 9th floor of the HEC building to investigate the appropriate file from the Records Section. The ability to quickly peruse a number of records is impracticable under the present system

C 1.1.8 Inefficiencies of the Central Records System

It is often difficult to anticipate the index reference that Central Records Section staff have assigned to a file thus searching may become unnecessarily complicated.

This seems largely due to the lack of cross references in their filing system. Inconsistencies in the geographical location index are another cause of unnecessary search delays.

C 1.1.9 Inadequate Reference System to Commission Land

This is in reference to the operation of the LAS Deed Card Index. The difficulty with this system is finding a card when the precise property locality is unknown, or the locality name used in the index is unknown. This problem is confined to personnel who use the system less frequently than LAS staff.

C 1.1.10 Deficiencies of the Wayleave Register

There are many specific difficulties in searching wayleaves. The greatest of these is associating a current title with a wayleave agreement negotiated, perhaps fifty years ago. Cancellation of title, transfer of ownership, subdivision of the original parcel, conversion from General Law to Real Property are all potential complications which could severely inhibit the

identification of the original agreement. If, however, certificate of title references and property owners were kept updated, these problems would largely disappear.

C 1.1.11 Access to Information Pertaining to All Land Activities Other Than Land Purchase.

The Deed Card Index is the only complete land index within the Survey Section, however, it is a reference to land purchases only. Complete indices for leases, licenses, other land interests and land disposal exist only through the Records Section ADEPT filing system which, as mentioned, has significant deficiencies.

C 1.1.12 Monitoring of Land Interests

Many Commission land interests are dynamic in nature. For example, some lease agreements come under review from time to time, the result of which could be an increased rental. The Commission may be unaware of this until served with a notice of the change. Monitoring such interests is too cumbersome to be administered manually. Doubt has been expressed as to the existence of Commission personnel able to give account of all property subject to leases. The branch most concerned with a parcel is often unaware of lease, or any other, conditions to which the land may be subject.

C 1.1.13 Confusion of Transmission Line Names

There are presently at least two methods of naming transmission lines, (i) source location to destination, and (ii) from turning point to turning point. The LAS use the former method.

C 1.1.14 Time Delays

This is in reference to the length of time taken to complete a job. Many job requests require months or perhaps years to complete and may be active for only a short period. At any time during that period, however, staff will be required to give information with regard to the progress of the request. Without an adequate filing system it is difficult for staff to remember

details of the request and its progress, especially if the job has been dormant for some time.

C 1.1.15 Demand for Land Information

Briefly, the problem pertains to the significant demand for land information by the Survey section, other Commission sections and the Public. Enquiries are submitted at a faster rate than can be processed.

C 1.1.16 Remoteness of Some Information Sources

Externally, for example, extensive searching is done at the Land Title Office (LTO). Locally, much searching is necessary in the Records section or in other branch files. This involves a considerable amount of time in transit between one office and another.

C 1.2 Conclusion

The outlined problems are, in some cases, local to the Survey section, others are caused by external factors originating from various Commission branches. It is suggested that the principal reason for the persistence of the latter in conflicting political forces within the Commission compromising management.

The Commission consists of five branches, each with its own infrastructure and Head, who is responsible to the Commissioner. All branches have interests and installations situate on land and assume dominion over these in all matters, including land policy. The ability of the LAS to formulate and administer Commission land policy is, therefore, often compromised by other branches wishing to administer a policy of their own. Certainly, the fact that all land related correspondence does not pass through the LAS is a hindrance to the administrative ability of that section.

It is observed that it is not the task of the LIS project to solve political problems, rather to acknowledge their existence and successfully operate within the framework they create.

Many problems local to the Survey section have not been previously

overcome since they have not been recognised as existing. For example, a simple method of indexing files to some staff members may present extreme difficulties to others. Those who find the system easy to use do not recognise that there is a problem. Such conflicts can, and do, frequently occur, often due to a different perspective of the data. For example, the LAS index land parcels using a locality code which usually corresponds to the municipality name. Drafting section prefer to work with map references, coordinates or survey diagram references.

Every avenue has been pursued to identify problems within the Survey, LAS and other sections. It is held that all significant deficiencies have been identified, and any further study in this area would merely reveal minor irritations rather than serious problems.

Chapter 2. IDEAL SYSTEM CHARACTERISTICS

This section introduces the characteristics of an ideal information system, that is, one that can process and provide an information product. This question was raised in both questionnaires and interview sessions with external users. Its purpose is to establish all present and foreseeable future requirements of the system, whether realizable, or not. This provides a complete picture of the type and extent of information and system characteristics required.

It is accepted that some requirements may not be satisfied given the resources available. In this case the report serves the purpose of documenting the nature of the requirements for use in a future project.

C 2.1 The ideal system characteristics are described as follows,

C 2.1.1 Simple access to data, simple operation

A prime objective of the system is to be accessible to all users. In general, any member of the Survey section requiring land related information should be able to retrieve it using the LIS. To realise this the system must be simple in operation for the most inexperienced user, yet fast and sophisticated for those using the system continuously.

C 2.1.2 Completeness of data - "one stop system" concept

The one stop system enables rapid access to land parcel records which contain all salient data relating to a parcel and the references of the associated ADEPT file documents. This is operative for probably 90% of cases, others may have special conditions, possibly too involved to record in the computer system. The existence of non-standard conditions will be indicated against the record and reference made to the source documents containing the particular conditions.

C 2.1.3 Automatic cross referencing

Cross referencing between various Commission indices is automatically updated.

C 2.1.4 Graphical display

Storing parcel and transmission tower coordinates will enable spatial location to become an index into the system as well as providing the basis for graphical display. Coordinate storage will facilitate some extent of spatial analysis.

Graphical display can be approached at two levels, namely:

- (a) Enquiries dealing with a single parcel.

The parcel boundary and features are displayed, indicating portions subject to lease or recently sold, transmission line traverses, etc. Neighbouring parcel boundaries might also be shown.

- (b) Wayleave related enquiries.

A portion, or the entirety, of a transmission line is displayed detailing all parcel along the corridor.

C 2.1.5 Automatic updating of certain data (parcel owner and C.T.)

This discussion raised the possibility of updating parcel owner names from the State Government VALTAX system and certificate of title references from Lands Department computer printout of PID numbers and title references for 1:25,000 series map sheets. Unfortunately, update of C.T's for PID's occurs only with map revision (5 years). These issues are being investigated with staff from the respective organisations.

C 2.1.6 Monitoring job requests

This includes the following tasks;

- (i) Periodic listing of outstanding jobs,
- (ii) Listing high priority job requests,
- (iii) Listing jobs with plans completed but not lodged,
- (iv) Listing jobs not completed in a given time or by a given date.

C 2.1.7 Job request analysis to produce workload statistics

The job inventory requires an analytical component with the capability to monitor the database and produce a report, for example, detailing:

- (i) the total number of jobs in the system
- (ii) the number of each job classification
- (iii) the total number of jobs processed during the last period
- (iv) the number of each job classification processed in the past period
- (v) the total number of job requests submitted in the past period

C 2.1.8 Further cross referencing

Further cross referencing for deed packets. For example, access could be gained directly from a 1:25,000 map or county chart, title reference, name of the property owner, address, coordinates, etc. The Drafting section has a limited PID system for large, ill-defined parcels which would be a useful cross reference.

C 2.1.9 Aerial photography

Referencing parcels to aerial photography coverage. This could be achieved by coordinating all flight path corridors. The photo coverage for a parcel is then determined by comparing its coordinates with those of the flight path corridors.

Another possibility is to photo-digitise the Flight Plan Folder (1:100,000 map sheets with the aerial photography coverage shaded). This would present a visual display of the flight path from which the position (and coverage) of the parcel could be interpreted.

Chapter 3. SYSTEM CHARACTERISTICS

The operational Land Information System can be conceptualised as two cooperating individual functional units, namely:

- (i) the basic data-management system
- (ii) specific applications and enquiry information system.

The basic data-management system is the nucleus of the LIS. It is responsible for entering and modifying data in the system. It is not capable of, nor is it intended to, manipulate the data in any way other than present it to the user, allow it to be modified, then store it again in an efficient manner.

The application and enquiry system is different. Its purpose is to provide information rather than mere data. This is achieved by combining various data which are then interpreted by the user as information. Combining the same data in a different manner yields different information. The information, per se, is not stored in the system, but is interpreted from data that is stored, i.e. it is derived information. Thus, the system manipulates data in order to provide users with an information management utility capable of decision support.

In practical terms, the basic data-management system is used to maintain the system, enter new parcel data, update existing parcels, retrieve and display parcel records given a single identifier (e.g. UPI or address). The application and enquiry system is intended to retrieve records on the basis of complicated criteria evaluation. For example, retrieve the parcel(s) on the transmission line between Parmeston and Trevalyn where the line has been replaced and the new easement width is 3m wider. As a result of the enquiry you require only the parcel address and the owner's name.

Monitoring of lease agreement expiration or review dates with the intention of issuing warnings some time prior to the event is an example of the application system.

The intention of the questionnaire and interviews is to reveal areas in the operation of the Survey and LA sections where the forementioned computer application and enquiry system is appropriate.

C 3.1 The following applications have been identified:

c 3.1.1 Periodic listing of outstanding job requests

In order to keep track of job requests submitted, but not completed, the system will produce, on a given date of each month, or at the user's request, a report of all such jobs with the relevant statistics regarding the number of jobs and their nature.

C 3.1.2 Listing of high priority requests

At the user's request, a report of all high, or any other specified, priority is produced with the relevant statistics.

C 3.1.3 Listing of all plans completed but not lodged

At the user's request a report is produced detailing all survey requests with completed, but not lodged, survey plans.

C 3.1.4 Listing of all requests not completed by the required date of completion

At the user's request a report is produced detailing all survey requests that are still active after their requested completion date.

C 3.1.5 Job processing analysis

On a given date of each month, or at the user's request, the information system is automatically interrogated, analysed and a report produced which details a current and quantitative view of the requests on hand, in progress and recently completed.

C 3.1.6 Perusing records

Provide the ability to peruse parcel records, or specified portions thereof, under user specified retrieval criteria.

C 3.1.7 Graphical display

Since parcel coordinate data is intended to be captured in the system, a form of graphical display is desirable. Display of individual parcels could be provided in the initial system although this would not support on screen graphical data editing.

C 3.2 Extended applications (not intended as part of the initial system)

C 3.2.1 Monitor rent review and expiration

The system continually monitors the status of rent reviews and expiration dates. Warning notices are printed at specified intervals prior to the event date.

C 3.2.2 Wayleave enquiry intelligence

This application performs a search for the existence of a wayleave over a subject parcel. The computer is given all known parcel identification from which it seeks the appropriate wayleave agreement. If existent, a letter is prepared for the enquirer informing him of the details of the agreement.

Chapter 4. DATA DEFINITION TABLE

Potential data considered for inclusion in the system have been identified through three principal sources, namely:

- (i) questionnaire and follow up interview sessions,
- (ii) Familiarisation exercise, (Appendix A)
- (iii) partial list of data items submitted by Chief Surveyor.

The following table list all identified data items.



DATA DEFINITION TABLE

ITEM NAME	DESCRIPTION	SOURCE	FREQUENCY OF UPDATE	EVENT	STORAGE LOCATION	IMPORTANCE RATING	TYPE	SIZE	ORDER	INDEX
EASTING NORTHING	A.M.G. COORDINATES	SURVEY DIAGRAM OR 1:25000 MAP	-	PARCEL SUBDIV. " CONSOLIDATION LAND ACQUISIT. ADDING A FEATURE	1:25000 MAP SHEET		N	2-3	-	✓
STREET NO. " SUFFIX " NAME " SUFFIX SUBURB/TOWN POSTCODE	URBAN ADDRESS ITEMS	MISCELLANEOUS CORRESPONDENCE	PROBABLY NOT UPDATED	-	LAS DEED BOX		N A/N A A A N	2 1 4 25 25 2	✓	✓
RSD NUMBER LOCALITY	RURAL ADDRESS		PROBABLY NOT	-			N A	2 25	✓	✓
PARISH MUNICIPALITY			PROBABLY NOT	-	DEED PACKET W/L AGREEMENT	LOW HIGHER	A A	25 25	✓ ✓	
PROPERTY/ INST. NAME	PROPERTY/INSTALL- ATION NAME	-	-	-	DEED BOX		A	60	✓	✓
PROPERTY/ INST. REF#	PROPERTY/INSTALL- ATION REF. NUMBER	-	-	-	RATE BOX		A/N	10	✓	✓
U.P.I.	LANDS DEPARTMENT U.P.I.	LAND DEPT. 1:25000 MAP SHT. SERIES	5 YEARS	MAP REVISION	-		N	3	✓	✓
MAP SERIES SHEET NAME SHEET REF.NO.	MAP REFERENCE	MAINLY LANDS COULD BE ANY ORGANISATION	-	-			A A N	35 25 2	✓	✓

DATA DEFINITION TABLE

ITEM NAME	DESCRIPTION	SOURCE	FREQUENCY OF UPDATE	EVENT	STORAGE LOCATION	IMPORTANCE RATING	TYPE	SIZE	ORDER	INDEX
NAME	NAME OF ORIGINATOR	REQUEST PROFORMA	-	-	SURVEYS L.A.S.		A	40 or 12	A-Z ?	-
TITLE	TITLE OF ORIGINATOR TITLE OF BRANCH/ SECTION HEAD	"	UPDATED IN L.I.S.		"		A	40 or 12	-	-
ADDRESS	ORIGINATOR'S ADDRESS	"	-	-	"		A	60	-	-
LAS FILE	LAS FILE REF. - INDICATION OF EXIST- ENCE & LAS FILE	"	-	-	"		CODE	1	-	-
COMMISSION FILE	ADEPT REFERENCE	"	-	-	"	HIGH	A	12	A-Z	✓
JOB #	JOB REF. NUMBER	"	-	-	"		A	12	A-Z	-
JOB NAME	JOB TITLE	"	-	-	"		A	60	A-Z	✓
DESCRIPTION	JOB DESCRIPTION	"	-	-	"		A	240	-	-
ALLOCATION	PERSON ALLOCATION WORK	"	-	-	"	HIGH	A	25	-	-
DATE	DATE REQUEST REC'D	"	-	-	"		DATE	8	A-Z	-
COMPLETION DATE	REQUESTED COMPLETION DATE	"	-	-	"		DATE	8	A-Z	-
PRIORITY	PRIORITY OF COMMENCEMENT	"	-	-	"		CODE	1	A-Z	-

DATA DEFINITION TABLE

ITEM NAME	DESCRIPTION	SOURCE	FREQUENCY OF UPDATE	EVENT	STORAGE LOCATION	IMPORTANCE RATING	TYPE	SIZE	ORDER	INDEX
FLIGHT #	AERIAL PHOTOGRAPHY FLIGHT NUMBER (=film number)	FLIGHT PLAN FOLDER	-	-	MAP REPRO. SECTION		I ?	2	A-Z	✓
PHOTO #	PHOTOGRAPH # UNIQUE TO 2 CERTAIN FILM (FLIGHT)	"	-	-	"		I	3	A-Z	✓
SCALE	SCALE OF PHOTOGRAPHY	"	-	-	"		A	10	-	-
FILE REF.	SURVEY FILE FIELD BOOK/PAGE # REF.	SURVEY FILE	-	-	DRAFTING SECTION		A	10	A-Z ?	✓
DRAWING REF.	REFERENCE OF THE FINAL REGISTERED DRAWING		-	-	"		A	10	-	✓
SURNAME	PROPERTY OWNER'S NAME	DEED PACKETS TITLES OFFICE	AVERAGE ONCE IN 8 YEARS	TRANSFER OF OWNERSHIP	RECORDS FILE DEED PACKET		A	25	A-Z	✓
INITIALS	4 INITIALS	VALUATION BR.					A	24		
"	ADJACENT OWNER'S NAME		"	"	"		A	3		
VOLUME FOLIO	ADJACENT OWNER'S C/T	"	"	"	"		I I	2 2	A-Z A-Z	✓ ✓
CURRENT STATUS	STAGE OF JOB COMPLETION	PERSON RESP. FOR JOBS	1 DAY TO 2 YEARS	COMPLETION OF STAGE	SURVEY SECT. LAS		A	30	-	-
STATUS DATE	DATE CURRENT STATUS WAS REACHED	"	"	"	LEGAL BRANCH		DATE	8	-	-

DATA DEFINITION TABLE

ITEM NAME	DESCRIPTION	SOURCE	FREQUENCY OF UPDATE	EVENT	STORAGE LOCATION	IMPORTANCE RATING	TYPE	SIZE	ORDER	INDEX
MAINTENANCE	ROAD MAINTENANCE COST & DESCRIPTION	INTER-BRANCH FILE	-	-			A/\$	30	-	-
ROAD EXPENSES	RENTAL FOR ROAD	RECORDS FILE	-	-			A/\$	30	-	-
TOLL	DETAILS OF ROAD TOLL	RECORDS FILE	-	-			A/\$	30	-	-
PERMITS	DETAILS OF PERMITS	RECORDS FILE	-	-			A	60	-	-
TOWER NUMBER	TOWER REF. NUMBER	INTER-BRANCH FILE	-	REMOVAL OF OR REPLACEMENT OF TOWER			A/N	10 ?	A-Z	✓
EASTING	TOWER CO-ORDINATES	SURVEY DRAW.	-	"	SURVEY SECTION		I	3	A-Z	✓
NORTHING							I	3		
POLE LINE	INDICATION OF PRIVATE OR FEEDER LINE	DIST. BRANCH RECORDS FILE	-	-			CODE	1	-	-
RATE #	VALUATION BRANCH RATE #	VALTAX	5 YEAR	5 YEAR REVALUATION	STATE TREASURY VALUATION BR.		I	2	A-Z	✓
AMOUNT	AMOUNT OF RATES PAID	VALTAX	5 YEAR	"	"		\$	3	A-Z	-
SERVICES	SERVICES IN LIEU	VALTAX	-	CHANGE OF SERVICES	"		CODE	1	-	-

DATA DEFINITION TABLE

ITEM NAME	DESCRIPTION	SOURCE	FREQUENCY OF UPDATE	EVENT	STORAGE LOCATION	IMPORTANCE RATING	TYPE	SIZE	ORDER	INDEX
AREA	AREA OF LEASE (HA)	CORRESPOND. FILE, LAS LEASE INDEX SURVEY DIAG.	-	-	RECORDS SECTION L.A.S.		I	3	A-Z	-
LESSEE/ LESSOR	NAME OF LESSEE OR LESSOR	RECORDS FILE	-	EXPIRATION OF LEASE	"		A	25	A-Z	✓
DESCRIPTION	PHYSICAL DESCRIPTION OR PROPERTY	INTER-BRANCH FILE	-	-		LOW	A	30	-	-
PUBLIC ROAD RESERVE	INDICATION OF SUB-STATION RESIDENT ON PUBLIC RD. RES.	RECORDS FILE PUBLIC FILE	-	-	RECORDS SECT. OTHER GOVT. DEPTS.	QUITE HIGH	CODE	1	-	-
STRUCTURE	DESCRIPTION OF STRUCTURES ON THE PARCEL	INTER-BRANCH FILE	-	-		QUITE HIGH	A/N	60+	-	-
PURCHASE DATE	DATE OF PURCHASE OF PARCEL	RECORDS FILE	-	-	RECORDS SECT.		DATE	8	A-Z	-
AREA	AREA OF PARCEL (HA)		-	-			N	3	A-Z	-
LAND USE VEGETATION	DESCRIPTION OF LAND USE & VEGETATION ALONG W/L LINE	INTER-BRANCH FILE	-	-		HIGH	A/N	60-100+	-	-
NAME	NAME OF ROAD	PUBLIC FILES	-	-	OTHER STATE GOVERNMENT DEPARTMENTS		A	25	A-Z	-
WIDTH	WIDTH OF ROAD OPERATIONAL		-	-			R	1	-	-
WIDTH PAVEMENT	RESERVE DESCRIPT.OF PAVEMENT TYPE OF ROAD		-	-			A	25	-	-

DATA DEFINITION TABLE

ITEM NAME	DESCRIPTION	SOURCE	FREQUENCY OF UPDATE	EVENT	STORAGE LOCATION	IMPORTANCE RATING	TYPE	SIZE	ORDER	INDEX
DEED NUMBER		DEED PACKET	-	SUBDIVISION CONSOLIDATION	DEED PACKET		A/N	10?	A-Z	✓
CONVEYANCE #		DEED PACKET	-							
CONVEYANCE #		DEED PACKET	-	"	"		A/N	10?	A-Z	✓
VOLUME	C.T. REFERENCE	DEED PACKET	-	"	"		N	4	A-Z	✓
FOLIO							N	3		
SURVEY PLAN #	SURVEY PLAN/DIAGRAM REF.	TITLE, DEED, SURVEYS FILE	-	"	"		A/N	10	A-Z	✓
DATE OF REG.	REG. DATE OF INSTRUMENT	INSTRUMENT		"	"		DATE	8	A-Z	✓
WAYLEAVE CHANGE	DESCRIPTION OF CHANGE TO WAYLEAVE			WAYLEAVE CHANGE	WAYLEAVE AGREEMENT	HIGH	A/N	60	-	-
DATE OF CHANGE	DATE OF CHANGE OF WAYLEAVE			"	RECORDS FILE		DATE	8	A-Z	-
ENCROACHMENT INDICATION	INDICATION OF ENCROACHMENT TO W/L & BRIEF DESCRIPTION	ELECTRICAL BRANCH - TRANS. LINES	4 MONTHS	WAYLEAVE INSPECTION	ELECTRICAL BRANCH ?	HIGH	A/N	100-200	-	-
LAST DATE OF INSPECTION	INSPECTION DATE OF WAYLEAVE		"	"	"		DATE	8	A-Z	-
CONSIDERATION	CONSIDERATION FOR WAYLEAVE	WAYLEAVE AGREEMENT	STATIC	-	WAYLEAVE AGREEMENT		\$	3	-	-
DIMENSIONS										
WIDTH	EASEMENT WIDTH	"	-	MAJOR LINE			R	2	-	-
DEPTH	DEPTH			CHANGE OR	"		R	1	-	-
LENGTH	LENGTH			REPLACEMENT			R	1	-	-

DATA DEFINITION TABLE

ITEM NAME	DESCRIPTION	SOURCE	FREQUENCY OF UPDATE	EVENT	STORAGE LOCATION	IMPORTANCE RATING	TYPE	SIZE	ORDER	INDEX
EASEMENT DESCRIPTION PURPOSE	TYPE OF EASEMENT: W/L, DRAINAGE ETC. & A BRIEF DESCRIPTION IF APPLICABLE	TITLE WAYLEAVE AGREEMENT	-	ADDITION TO SERVICE	DEED PACKET	HIGH	A	60-100	-	-
REG. DATE	DATE OF REGISTER OF ENCUMBRANCE	TITLE	-	-	DEED PACKET		DATE	8	A-Z	-
SPECIAL CONDITION	INDICATION OF SPEC. CONDITIONS OF THE ENCUMBRANCE BINDING ON THE COMMISSION	RECORDS FILE	-	-	RECORDS FILE DEED PACKET		A/N	30	-	-
MAINTENANCE AGREEMENT	INDICATION IF EXISTANCE OF A MAINTENANCE AGREEMENT FOR THE ENCUMBRANCE	"	-	-	DISTRIBUTION BRANCH	LOW	CODE	1	-	-
ACTION DATE	DATE ON WHICH ACTION IS REQUIRED AS INDICATED BY SPECIAL CONDITIONS OR PERIOD BETWEEN ACTION DATES	"	-	-	DEED PACKET RECORDS FILE		DATE	8	A-Z	-
RENTAL	FOR EASEMENTS	RECORDS FILE AGREEMENT		RENT REVIEW	"		\$	2	-	-
COMMENCEMENT DATE	COMMENCEMENT DATE FOR A LEASE AGREEMENT	"	STATIC	-	"		DATE	8	A-Z	-
DURATION	DURATION OF LEASE	"	STATIC	-	"		R	1	-	-
EXPIRATION	EXPIRATION DATE OF LEASE	"	STATIC	-	"		DATE	8	A-Z	-

DATA DEFINITION TABLE

ITEM NAME	DESCRIPTION	SOURCE	FREQUENCY OF UPDATE	EVENT	STORAGE LOCATION	IMPORTANCE RATING	TYPE	SIZE	ORDER	INDEX
REVIEW	DATE WHICH LEASE IS SUBJECT TO LEASE REVIEW OR PERIOD BETWEEN REVIEWS	RECORDS FILE	-	-	DEED PACKET RECORDS FILE		DATE	8	A-Z	-

Chapter 5. ACCEPTANCE CRITERIA

Formal project control and monitoring during implementation is necessary to ensure that the development adheres to the design specifications. This is again reviewed upon completion of the project by comparing the operational system with a set of user defined acceptance criteria or success factors.

These criteria are solicited from the users during the requirements analysis and formulation stage. Obviously, it is difficult to determine such criteria at the outset of the project. It is easier to wait and see what can be done then decide whether it is sufficient, especially if the users have little experience with computer based information systems.

The following acceptance criteria originated, in the main, from one interviewee, as most regarded any system that would overcome their particular problems (see chapter 1 of this report) as a fair indicator of success.

C 5.1 Acceptance Criteria

C 5.1.1 Work Performance

It is expected that the work performance of those using the system will be increased. A quantitative target of 40% has been suggested.

C 5.1.2 Complaints and Bouquets

Unfortunately, it is presently common to receive complaints from other sections regarding the progress of work requested. This is especially so since the Survey and LA sections are at the forefront of Commission development. Typically, all work relating to land cannot proceed until either an investigation survey is completed or the necessary administrative land procedures have been completed. The success or otherwise of the system may be assessed by the feedback from other sections.

C 5.1.3 Personal Disposition and Relief of Work Tension

An evaluation of whether use of the operational LIS has improved the personal disposition of staff and provided some relief from work pressure.

C 5.1.4 Staff Attitudes

It is observed that staff whose workload increases regardless of their personal efficiency, become demoralised. This in turn will ultimately impair their efficiency and a downward spiral results. Although job satisfaction is difficult to measure, it is intended that increased job satisfaction should be a success indicator.

C 5.1.5 System Performance

System performance is qualified by ease of use, reliability, response time and relevance of enquiries and applications to the existing information problems. Quantifying these factors is more difficult, however, the system would be expected to rate better than "fair" on the following scale:

Poor
Fair
Good
Excellent

C 5.1.6 Level of Adoption

The final, and perhaps, most conclusive success factor is the extent to which the staff adopt and rely on the system. Obviously only a high level of adoption will enable the above criteria to be satisfied. It is observed that such a level of adoption will be achieved only if staff have confidence in both the computer system, per se, and the encapsulated data. This implies characteristics of system reliability, performance, quality of user interface, data integrity, etc.

C 5.2 Conclusion

The above represent a comprehensive range of criteria and, in some cases, are very ambitious. Furthermore, many factors will depend on broader issues, such as the success and extent of user education, cooperation, willingness to learn and preparedness to develop new work skills and procedures. Some criteria, although difficult to quantify, will be readily identifiable however in the form of a litmus test.

Chapter 6. IMPLEMENTATION PROPOSAL

The information problem experienced by the Survey section is shared by many other Commission sections. In many of these cases, the problem is also in relation to land information. The object of the current contract is to provide a land information system for the Survey section, however, it must be realised that the problem extends beyond this artificial boundary. In fact, there may not be a boundary to this problem. It is prevalent throughout the Commission. The requirements analysis and formulation study has directed attention to the general Commission land information problem as well as concentrating on the specific needs of the Survey section. On the basis of this it is estimated that the proposed system for the Survey section would accommodate approximately 70%-80% of the initial requirements of the other sections. The problem then becomes one of providing access to the system for those section.

The foregoing chapters have described the problems confronting the Survey section with regard to land information. The chapters have also described applications that the system will be required to perform, defined the data items to be included in the system and isolated criteria by which acceptance of the system can be judged.

This chapter describes the system implementation issues. These are concerned with the arrangements necessary to transform the LIS from a design system to an operational information system.

C 6.1 Access to the System

C 6.1.1 Public Scenario

The LIS is available to anyone in the section requiring land information. Authorised personnel from other section may also have access. The system is operated by whoever requires the information and maintenance (data entry, update, etc) is performed on the same basis.

C 6.1.2 Information Centre Scenario

This represents a controlled, centralised approach. A Systems Administrator is appointed with a part time staff. The task of the Administrator is to prioritise and schedule the processing of information requests and data maintenance. The Administrator and support staff are responsible for processing all computer work, including LIS, JIS and MOSS. Information products are in hard copy form and could be deposited for collection or distributed directly to the people concerned.

C 6.2 Advocacy Planning

The success of the public scenario is endangered by its uncontrolled nature. The result of personnel wishing to access the system on an ad-hoc basis will result in problems, namely

- (i) two or more persons wishing to access the system simultaneously,
- (ii) some people will be very slow in operating the system and thus hold up others,
- (iii) increasing probability of :
 - (a) corrupting data
 - (b) "crashing" the system
 - (c) damaging hardware.

The advantage of this scenario is in the achievement of a fundamental project objective, namely, to provide simple access to land information for all staff.

The strength of the Information Centre scenario lies in its organised structure. The described personnel arrangement enables a small number of staff to become highly proficient in the use of the system. The same staff could operate the other computer applications, MOSS and JIS. They would quickly become expert in processing all types of computer related operations thus enhancing the efficiency of the entire section. The risks to the system are considerably less since only experienced, authorised personnel can use it.

This applies equally to data integrity as much as to physical damage to the computer.

The weakness of the approach, however, lies also in its organised structure. In particular, the centralised control. Several problems could emerge, for example:

(i) the expertise of operating the computer system is limited to a small number of people. This simply duplicates an already existing problem. If they were to leave, become ill, etc, then the usefulness of the computer is impaired.

(ii) the result of an information request is in the form of a computer print-out which the originator may receive, an hour, or a day following the submission of the enquiry. A more effective method for obtaining data is through an interactive, personal session with the information system. Frequently the information requested is not sufficient, i.e. one answer poses yet another question which results in a further request for information.

In fact, neither of the scenarios above would produce an entirely satisfactory result. The process of advocacy planning, however, provides the key to a solution by integrating desirable characteristics of both scenarios.

C 6.3 Proposed Organisation Structure

It is recommended that an Information Centre be established within the Survey section. Its brief being to administer, prioritise (under the guidance of the Chief Surveyor), and process the information requests from Survey and other sections. The Information Centre requires an Information Systems Administrator and a support staff of three. The duties of the Administrator are:

- (i) prioritise information requests in consultation with the Chief Surveyor,
- (ii) administer the processing of requests as determined by the priorities,

- (iii) perform basic maintenance operations on the hardware and software,
- (iv) participate in processing requests and providing information products,
- (v) take responsibility for administering the security and integrity of the LIS and JIS.

The support staff may be involved on a part time basis with the Information Centre and remain attached to their original area of the Survey section. Their brief is to perform the data entry, update and retrieval operations necessary to maintain the currency of the system and satisfy all information requests.

It is imperative that the Information Centre be directly responsible to the Chief Surveyor, with respect to information system work, rather than their immediate section leader. Thus the System Administrator can achieve information processing according to the set priorities without compromise.

To overcome the deficiencies of the centralised approach it is proposed that a second computer be made available for public access to the information system. This computer would be under the control of the Information Centre. Thus personnel wishing to browse many parcels or make enquiries likely to generate further enquiries, are able to hold interactive, personal sessions. This allows personnel other than the Information Centre staff to obtain experience in using the system, therefore broadening the potential base for relief staff. It will also provide direct access to land information for other sections as an alternative to submitting a request to the Information Centre. The second computer should be close to, but separated from, the main computer area. Thus public users would not interfere with the operation of the Information Centre, but will be close enough to seek expert advice, if required.

It is recommended that the system is operated initially with only a single machine until the actual information demand is determined. The method for installing any additional public access points can then be evaluated. (See Future Extensions).

C 6.4 Work Environment

The Information Centre will require a new work Environment. Fortuitously the Survey section is expanding to make use of the space vacated by the Geology section. This provides an excellent opportunity to physically establish the centre.

The work environment for intensive computer use must be cool, dry, reasonably quiet, dust free and removed from people traffic. These factors hold equally for machines and staff. Computers require a monitored cool, dry environment to avoid over-heating problems. Dust can cause serious problems if allowed to enter the machines. Like other work requiring intensive concentration, a reasonably quiet environment is essential. Passing people traffic is a disturbance and an unnecessary hazard. It simply increases the possibility of a machine being knocked and damaged. A greater risk, however, is that data being entered may be corrupted through such a knock or inadvertent tampering.

Computer operation places similar strains on the operator as does typing. Computer operators are therefore equally prone to discomfort and injury as typists and should therefore be provided with suitable chairs and tables to lessen these woes.

Concentration on a visual display unit (VDU) for protracted periods can lead to severe headaches resulting from eye stress. This can be eased if the operator has an avenue through which to, periodically, relax the eye muscles. A window outlet provides a perfect opportunity to do this.

C 6.5 Foreseeable Problems

There are potential problems in the proposed structure, although counter measures have been introduced. The establishment of an Information Centre could alienate staff from other staff not involved in the Centre. Staff jealousies and rivalry between personnel in the new Centre and those remaining in the existing sections could emerge. These would be most unfortunate side effects and could seriously undermine the operational ability of the new system were they to eventuate.

Careful, planned management can avoid these problems. Physical separation

of the Centre from other activities is as important as its unfortunate potential for creating alienation. However, installation of a public computer adjacent to the Information Centre is intended to minimise this feeling as well as provide the necessary public access to the information system. Part time membership of the Information Centre staff could resist feelings of jealousy or defensive negative attitudes from senior section members to losing their staff.

C 6.6 Prototype System

It is proposed that the formal system design be "frozen" upon completion. The LIS and application program code is then written and a prototype system developed. The process is not interrupted in the event of discovery of a design error (unless it is essential). The prototype is evaluated and all evident "bugs", potential improvements, data omissions and design errors are reported. A second iteration proceeds based on the evaluation report.

C 6.7 Future Extensions

The nature of future extensions to the system will depend on both the demand for information and the advancement of technology. There are three extensions possible in the short term, namely:

(i) Duplicated Databases

The LIS database can be duplicated on another computer. All data update is restricted to the master machine and a copy of this is used to update the slave database at regular intervals.

Cost: \$6,000 to convert the floppy based Sirius to an LIS work station,
\$11,000 per additional work station (plus additional external hard disk @ \$4000.)

(ii) Time sharing System

The LIS program could be converted to run under a multi-user operating system without changing the hardware. Thus several users can access the same database simultaneously. In practice this would be

limited to a total of two, or three users.

Cost: \$9,000 using the floppy disk based Sirius.

(iii) Local Area Network

Up to eight users could share access to the database and printer through a computer (Sirius 10Mb) dedicated to serving the network.

Cost: \$7,000 for the first workstation, inc. floppy based Sirius,
\$5,000 for each additional workstation.

If the system is to service the land information system requirements of the Survey and other sections then one of the above configurations will be necessary.

Microcomputer technology is young and developing rapidly. It is likely that a more suitable or enhanced version of the above configuration will be available by the close of the year, 1984.

C 6.8 System Life-span

It is pertinent that the system should be assigned a life-span after which it should be replaced. This is not for reasons of paranoia regarding usage of leading technology. At some point in the future technology will have advanced to such an extent that, to remain with the present technology would be analogous to remaining with the manual file management system currently being replaced. The system could continue to operate after its life-span, however, the cost of so doing is the expense of not employing the latest information technology. The selected life-span is such that the estimated opportunity cost justifies the replacement of the system.

C 6.9 Recommendations

The following recommendations are therefore proposed:

- (a) the proposed system be regarded as version 1,

- (b) a five year life-span be adopted,
- (c) prior to the expiration of this period an investigation should be conducted into the application of appropriate new information technology to develop Version 2.

APPENDIX D

External Users and Distribution Branch

APPENDIX D. EXTERNAL USERS AND DISTRIBUTION BRANCH

D 1.1 External Users

Every Commission branch and section requires access to land information. The requirement in many cases is to geographically locate an item of land related information. Another fundamental reason for requiring land information is to determine the nature of Commission land interests and the interests, or encumbrances, of third parties.

Many requirements, however, are for technical data specific to a particular section. The data is usually related to land in that it pertains to an installation which is situate on land. This data is not considered land information in terms of the proposed system and is, therefore not included.

The scope of data selected for the proposed system is, however, quite broad and includes descriptions of:

- (a) Certificate of Title
- (b) Interests and encumbrances
- (c) Leases and licenses
- (d) Physical description of land
- (e) Structures built on land
- (f) Geographic location
- (g) Commission file references,
Survey file references, including survey drawings and aerial
photography
- (h) Encroachments on wayleaves
- (i) Rates
- (j) Transmission line towers
- (k) Changes to wayleave agreements
- (l) Survey and Land Administration job requests and progress

A series of meetings with potential external users (i.e. users from outside the Survey section) indicated that the system would be useful to their respective sections, even without specific technical data.

It is therefore suggested that the Survey section land information system will contribute toward easing the land information problem throughout the Commission. Nevertheless, three serious problems remain.

D 1.2 Technical Information System

There remains a desperate requirement among many sections for a "section specific technical" information system. The Distribution branch is a case in point. There are some 1/4 million power poles in Tasmania, each with a history and set of descriptive attributes regarding its construction, line carrying capacity, line status, geographic location and physical condition. There is an enormous quantity of data involved, too much to be efficiently managed by manual means. It is, nevertheless, very important data especially in relation to the maintenance of power poles and clearing of vegetation from power lines. The precedent for this was established following the Victorian bushfires.

D 1.3 Multi-User Access

As indicated, the system will be implemented on a single microcomputer, thus will not be conducive to multi-user access. Two options are possible,

- (i) "queue up" on the single user machine,
- (ii) expand the system to provide a form of un-sophisticated multi-user access.

Possible approaches to the latter are discussed in Chapter 6.4 "Future Extensions", of the Implementation report.

D 1.4 Incompatibility

This concerns the development of independent section specific information systems. It has been shown elsewhere that organisations, such as the HEC, introducing Management Information Systems (MIS) have allowed the development of these to take place independently within the organisation. The resulting systems are usually incompatible in terms of software, or hardware, or both.

Obviously, there is a need for both integrated land and technical information systems within the Commission. It may be desirable for some sections to marry land data and technical data in a single, fully integrated system while others would be satisfied with independent systems that are suitably cross referenced. Whatever future development transpires in the Commission with regard to section specific technical information systems, the data structure and design specifications of the Survey section LIS should be consulted in order to maintain compatibility between systems.

D 2. Distribution Branch Requirements

Close liaison and consultation has been maintained with the Distribution branch during the requirements analysis and formulation stage for the proposed LIS. Requirements from this branch have been included for consideration in the system with the following conclusions.

(i) Preliminary requirements were considered in consultation with the System Development Engineer (S.R.E. South) and subsequently documented in a memorandum to the S.R.E., dated 19th April 1984. Further discussions resulted in two additional data requirements appended to the list. Twenty eight data requirements were recorded.

(ii) The above requirements were considered by the Chief Surveyor and the consultant resulting in the adoption of 75% of the requirements. It was resolved that on the basis of restricted storage capacity certain data requirements could not be accommodated.

(iii) In terms of the original memorandum (19th April 1984) the rejected data items were:

(m) details of physical acres to easements

Power distribution details:

(p) reference numbers of distribution poles situate on property

- (q) date of erection of poles
- (r) date of last inspection
- (t) pole coordinates

Dynamic data

- (w) correspondence trail, i.e changes in a consumers wayleave.

The correspondence trails for fire hazard reduction and wayleave negotiations were also rejected for similar reasons.

All other items have been included.



APPLIED INFORMATION SYSTEMS PTY. LTD.

P.O. BOX 673, SANDY BAY, TAS. 7005 PHONE (002) 25 2082

19th April, 1984

Mr. Alan Sproule,
Systems Development Engineer (South),
Distribution Branch,
Hydro Electric Commission.

re: Preliminary Requirements for the Distribution Branch for Participation
in a Land Information System (LIS).

1. SCOPE OF INTEREST

The Distribution branch is interested in details of land holdings which are affected by Hydro Electric Commission (HEC) installations, namely:

- (i) minor substations
- (ii) regulator sites, wayleaves for overhead power lines and underground cables for lines of less than 22kV.

2. DETAILS OF DATA TO BE HELD

Property identification:

- (a) name of the current owner,
- (b) current C/T, deed or conveyance reference,
- (c) street address,
- (d) property boundary coordinates,
- (e) substation reference number.

Easement details:

- (f) name of the original owner (with whom the wayleave, lease, etc. agreement was negotiated),

- (g) C/T, deed or conveyance reference of the property at the time of the agreement,
- (h) date of the agreement,
- (i) dimensions of the easement, in particular width and length,
- (j) Records section (ADEPT) file reference,
- (k) expiration date of lease agreements,
- (l) indication of the existence of a maintenance agreement,
- (m) details of physical access to the easement - this could be best represented by graphical display on the VDU.
- (n) indication of the existence of special conditions included in the easement which are binding on the Commission,
- (o) indication of the existence of special conditions included in the easement which are binding on the Registered Proprietor.

Power distribution details

- (p) reference numbers of poles situate on the property,
- (q) date of erection of each pole,
- (r) date of last inspection,
- (s) depth of underground cables,
- (t) pole coordinates,
- (u) description of land in the easement, with particular reference to vegetation and land use. This is primarily for the purpose of fire hazard reduction clearing.

Dynamic data

- (v) indication of the status of negotiations re acquiring land or an interest in it,
- (w) correspondence trail re changes in a consumer's wayleave, e.g., for the purpose of upgrading supply.

Static data resulting from wayleave changes

- (x) description of the change,
- (y) date when the change was affected.

Property details

(z) indication of the rateability.

3. ACCESS TO THE INFORMATION SYSTEM

Direct access to the Survey Section Land Information System is desired for on-line updating of data and information retrieval. Such access is desired for the Southern and Northern regional Distribution Branches.

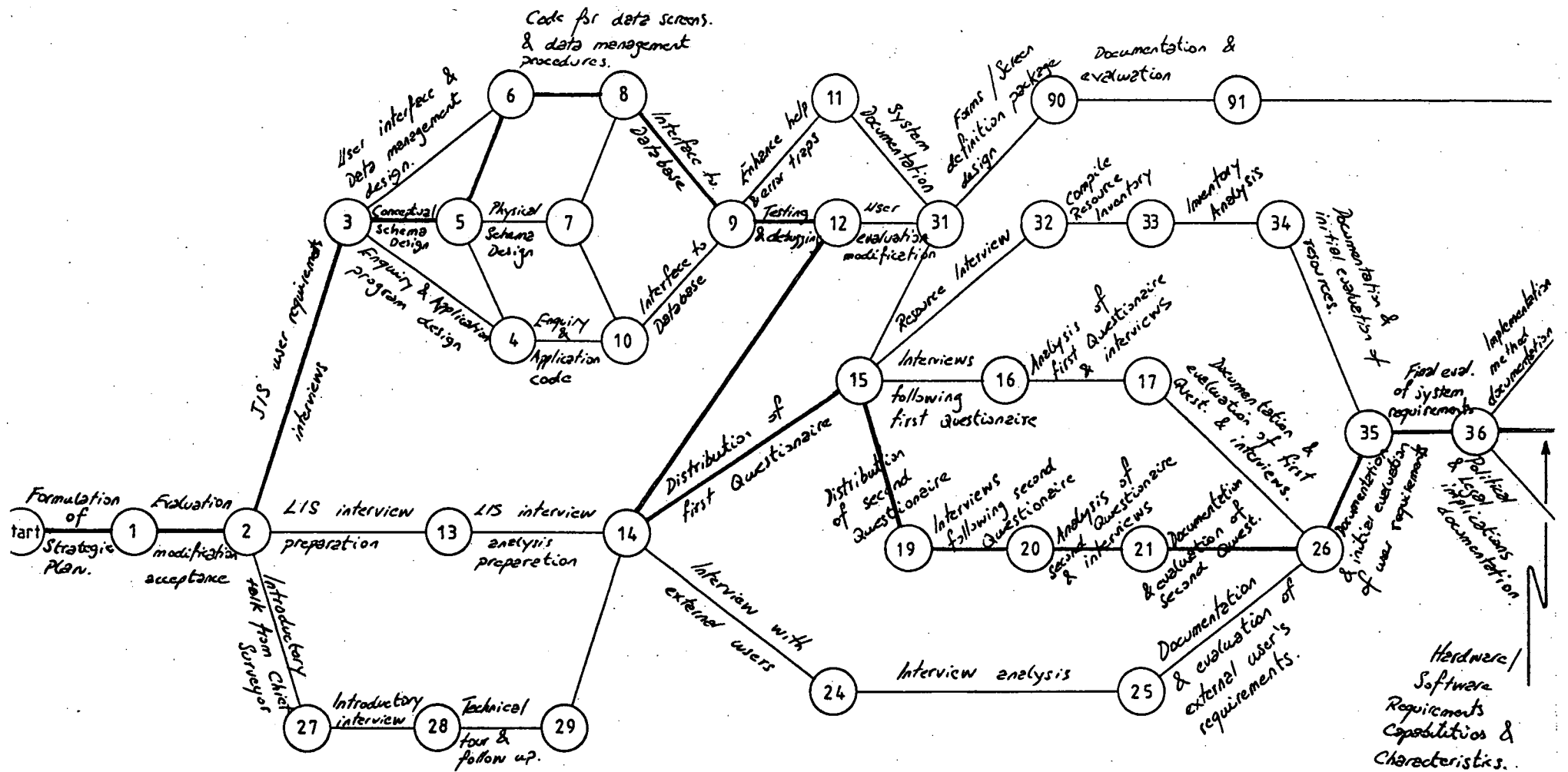
A handwritten signature in dark ink, appearing to read "W.R. Love". The signature is fluid and cursive, with a horizontal line underlining the name.

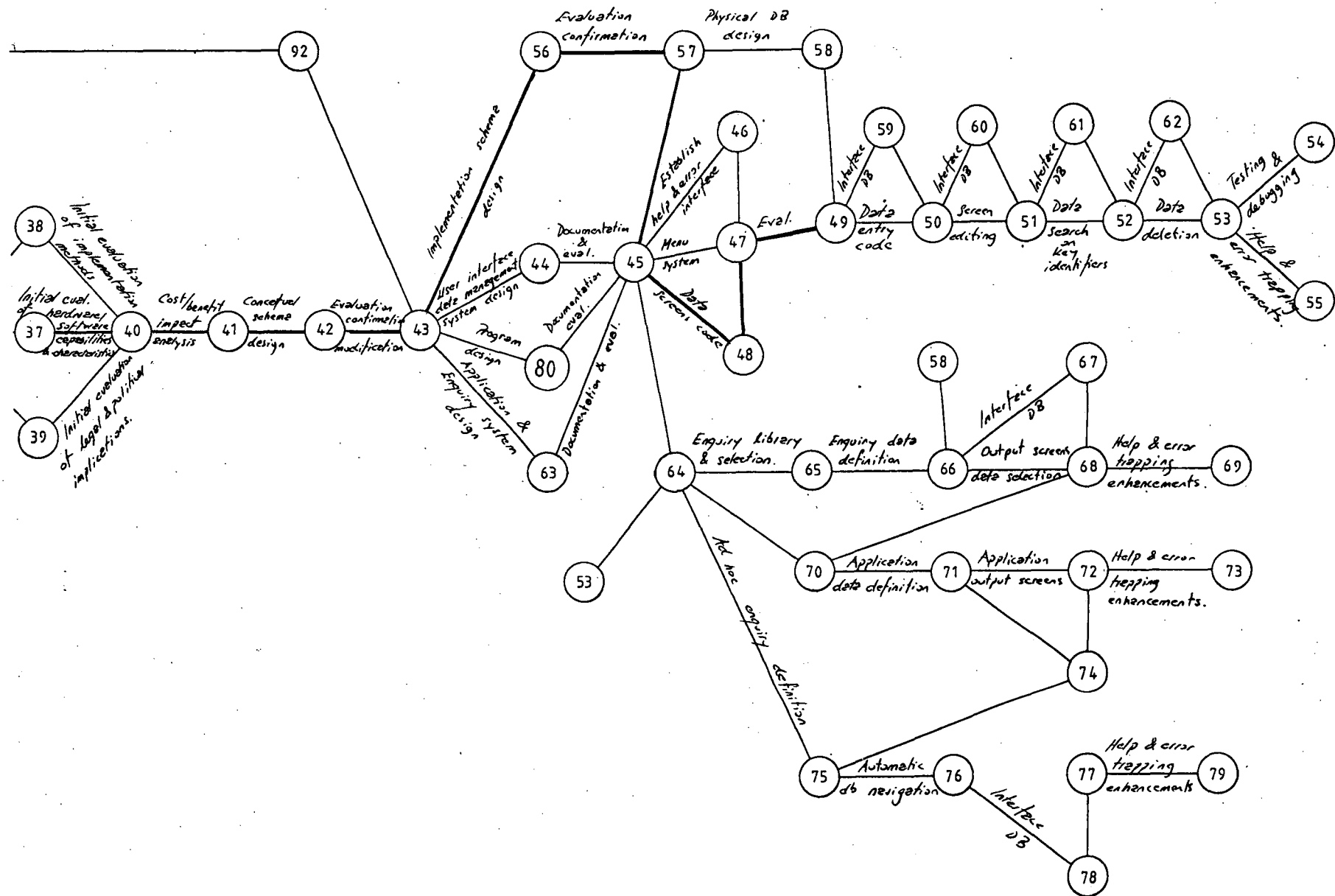
W.R. LOVE

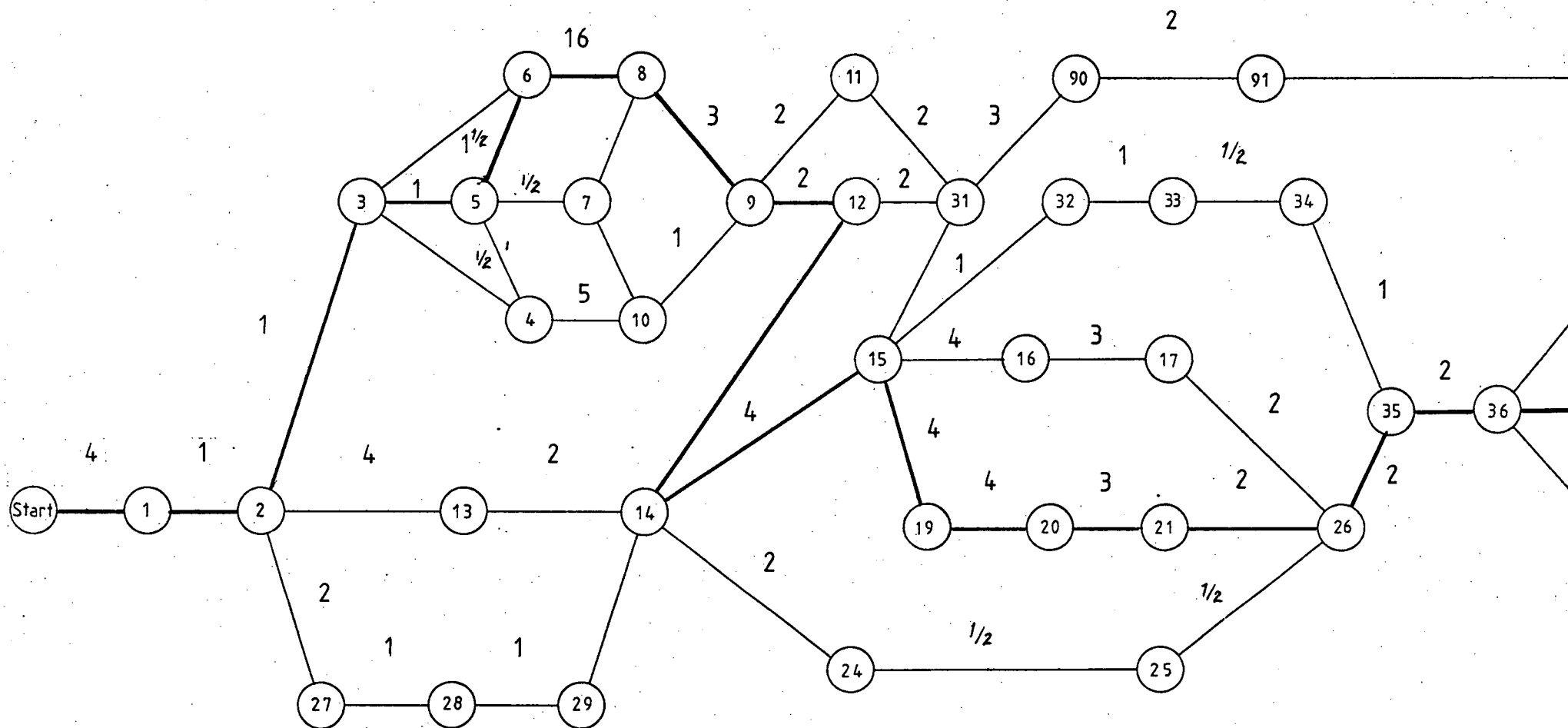
LIS SYSTEM DEVELOPER.

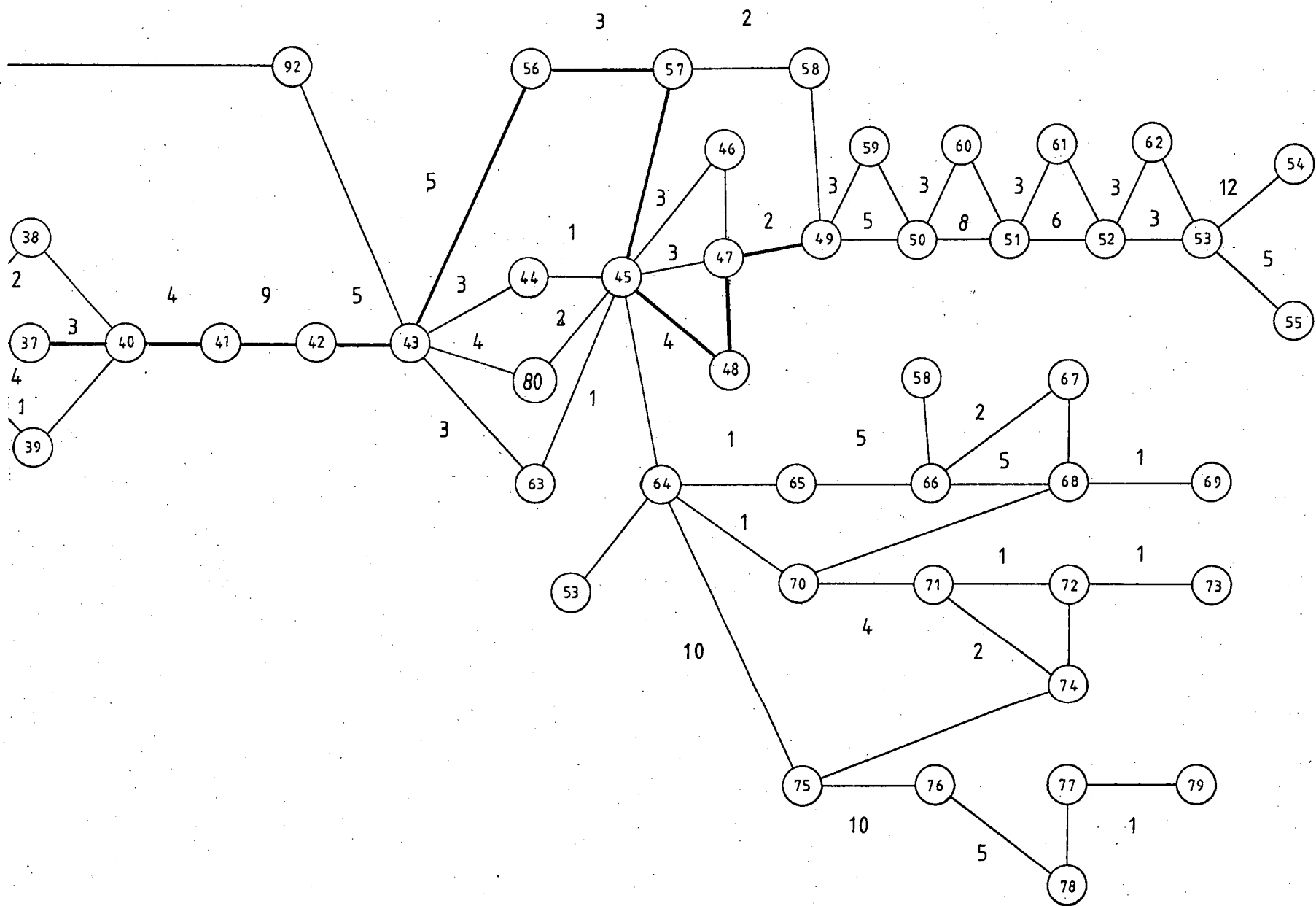
APPENDIX E

Activity Chart and Critical Path Diagram









APPENDIX F

A Microcomputer-based System for Managing Land Information

This article has been removed
for copyright or proprietary
reasons.

Love, W. R., 1985. A microcomputer-based
system for managing land information,
Australian surveyor, 32(7), 523-535